```
**********************************************************
    7419 Thu Jul 11 01:28:48 2013
new/usr/src/Makefile
onc plus-be-gone
first pass
**********************************************************
```

```
#
# 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 for system source
#
# include global definitions
include Makefile.master
#
# the Targetdirs file is the AT&T target.dirs file in a makefile format.
# it defines TARGETDIRS and ROOTDIRS.
include Targetdirs

COMMON_SUBDIRS= uts lib cmd ucblib ucbcmd psm man test
sparc_SUBDIRS= stand
i386_SUBDIRS= grub

#
# sparc needs to build stand before psm
#
$(SPARC_BLD)psm: stand

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

HDRSUBDIRS=     uts head lib cmd

# UCB headers are bug-for-bug compatible and not checkable against the header
# standards.
#
CHKHDRSUBDIRS=  head uts lib

#
# Headers that can be built in parallel
#
PARALLEL_HEADERS = sysheaders userheaders libheaders cmdheaders

#
```



```
# Directories that can be built in parallel
#
PARALLEL_DIRS = uts lib man

# The check target also causes smf(5) service manifests to be validated.
CHKMFSTSUBDIRS= cmd

MSGSUBDIRS=     cmd ucbcmd lib
DOMAINS= \
        SUNW_OST_ADMIN \
        SUNW_OST_NETRPC \
        SUNW_OST_OSCMD \
        SUNW_OST_OSLIB \
        SUNW_OST_UCBCMD \
        SUNW_OST_ZONEINFO

MSGDDIRS=       $(DOMAINS:%=$(MSGROOT)/%)
MSGDIRS=        $(MSGROOT) $(MSGDDIRS) $(MSGROOT)/LC_TIME

all :=                  TARGET= all
install :=              TARGET= install
all all_xmod :=         TARGET= all
install install_xmod := TARGET= install
install1 :=             TARGET= install
install2 :=             TARGET= install
install_h :=            TARGET= install_h
clean :=                TARGET= clean
clobber :=              TARGET= clobber
check :=                TARGET= check

.KEEP_STATE:

#
# Note: install does not cause a build in pkg.  To build packages,
#       cd pkg and do a 'make install'
#

all: mapfiles closedbins sgs .WAIT $(SUBDIRS) pkg

#
# The _msg build is a two-step process.  First, the _msg dependency
# causes recursive makes in $(MSGSUBDIRS), which stages raw message
# files in $(ROOT)/catalog.  Second, the action from the install
# target rule causes those messages to be post-processed from where
# they were staged in $(ROOT)/catalog, and the results placed into the
# proto area.
#
# The stage-licenses target causes the license files needed for
# packaging to be pulled from $(SRC) and $(CLOSED) and staged in
# $(ROOT)/licenses.
#
install: install1 install2 _msg stage-licenses
        @cd msg; pwd; $(MAKE) _msg
        @rm -rf "$(ROOT)/catalog"

stage-licenses: install2
        @cd pkg; pwd; $(MAKE) stage-licenses

install1: mapfiles closedbins sgs

install2: install1 $(SUBDIRS)

_msg: _msgdirs rootdirs install2 FRC
        @for m in $(MSGSUBDIRS); do \
                cd $$m; pwd; $(MAKE) _msg; cd ..; \
        done
```

```
mapfiles: bldtools
        @cd common/mapfiles; pwd; $(MAKE) install

clean clobber: $(SUBDIRS) head pkg

closedbins: bldtools $(ROOTDIRS) FRC
        @CLOSED_ROOT="$$ON_CLOSED_BINS/root_$(MACH)$${RELEASE_BUILD+-nd}"; \
        if [ "$$CLOSED_IS_PRESENT" = no ]; then \
                if [ ! -d "$$CLOSED_ROOT" ]; then \
                        $(ECHO) "Error: if closed sources are not present," \
                            "ON_CLOSED_BINS must point to closed binaries."; \
                        $(ECHO) "root_$(MACH)$${RELEASE_BUILD+-nd} is not" \
                            "present in $$ON_CLOSED_BINS."; \
                        exit 1; \
                fi; \
                $(ECHO) "Copying closed binaries from $$CLOSED_ROOT"; \
                (cd $$CLOSED_ROOT; \
                    $(TAR) cfX - $(CODEMGR_WS)/exception_lists/closed-bins .) | \
                    (cd $(ROOT); $(TAR) xBpf -); \
                ( cd $(ROOT); $(CTFSTRIP) $$(cd $$CLOSED_ROOT; $(FIND) \
                    ./kernel ./usr/kernel ./platform/*/kernel -type f -a -perm - \
                    $(EGREP) -vf $(CODEMGR_WS)/exception_lists/closed-bins) ); \
        fi

#
# Declare what parts can be built in parallel
# DUMMY at the end is used in case macro expansion produces an empty string to
# prevent everything going in parallel
#
.PARALLEL: $(PARALLEL_HEADERS) DUMMY
.PARALLEL: $(PARALLEL_DIRS) DUMMY

$(SUBDIRS) head pkg: FRC
        @cd $@; pwd; $(MAKE) $(TARGET)

# librpcsvc has a dependency on headers installed by
# userheaders, hence the .WAIT before libheaders.
sgs: rootdirs .WAIT sysheaders userheaders .WAIT \
        libheaders cmdheaders

#
# Top-level setup target to setup the development environment that includes
# headers, tools and generated mapfiles.  For open-only builds (i.e.: source
# trees w/o usr/closed), this also depends on the closedbins target (above)
# in order to properly seed the proto area.  Note, although the tools are
# dependent on a number of constant mapfiles, the tools themselves are
# required to build the generated mapfiles.
#
setup: closedbins bldtools sgs mapfiles

bldtools:
        @cd tools; pwd; $(MAKE) install

# /var/mail/:saved is a special case because of the colon in the name.
#
rootdirs: $(ROOTDIRS)
        $(INS) -d -m 775 $(ROOT)/var/mail/:saved

lint: FRC
        $(MAKE) -f Makefile.lint

_msgdirs:       $(MSGDIRS)

$(ROOTDIRS) $(MSGDIRS):
        $(INS.dir)
```



```
userheaders: FRC
        @cd head; pwd; $(MAKE) install_h

libheaders: bldtools
        @cd lib; pwd; $(MAKE) install_h

sysheaders: FRC
        @cd uts; pwd; $(MAKE) install_h

cmdheaders: FRC
        @cd cmd/fm; pwd; $(MAKE) install_h
        @cd cmd/mdb; pwd; $(MAKE) install_h

# each xmod target depends on a corresponding MACH-specific pseudotarget
# before doing common xmod work
#
all_xmod install_xmod: $$@_$(MACH)
        @cd uts/common/sys; pwd; $(MAKE) svvs_h

all_xmod_sparc install_xmod_sparc: FRC
        @cd uts/sparc; pwd; \
          $(MAKE) TARGET=$(TARGET) svvs pm wsdrv

all_xmod_i386 install_xmod_i386: FRC
        @cd uts/i86; pwd; $(MAKE) TARGET=$(TARGET) svvs

check:  $(CHKHDRSUBDIRS) $(CHKMFSTSUBDIRS)

#
# Cross-reference customization: skip all of the subdirectories that
# don't contain actual source code.
#
$(CLOSED_BUILD)XRDIRS += ../closed
XRPRUNE = pkg prototypes
XRPRUNE = pkg prototypes xmod
XRINCDIRS = uts/common head ucbhead
$(CLOSED_BUILD)XRINCDIRS = uts/common ../closed/uts/common head ucbhead

cscope.out tags: FRC
        $(XREF) -f -x $@

FRC:

# EXPORT DELETE START

XMOD_DELETE_FILES:sh = cat xmod/xmod_files

EXPORT_SRC:
        @cd $(CLOSED)/cmd/cmd-inet/usr.lib/in.iked;  pwd; $(MAKE) EXPORT_SRC
        @cd $(CLOSED)/cmd/cmd-inet/usr.lib/ike-certutils;  pwd; \
            $(MAKE) EXPORT_SRC
        @cd cmd/cmd-inet/usr.sbin; pwd; $(MAKE) EXPORT_SRC
        @cd $(CLOSED)/cmd/cmd-crypto/etc; pwd; $(MAKE) EXPORT_SRC
        @cd cmd/crypt;    pwd; $(MAKE) EXPORT_SRC
        @cd cmd/gss/gssd;         pwd; $(MAKE) EXPORT_SRC
        @cd cmd/krb5/kadmin;  pwd; $(MAKE) EXPORT_SRC
        @cd cmd/sendmail/src; pwd; $(MAKE) EXPORT_SRC
        @cd common/crypto/aes;    pwd; $(MAKE) EXPORT_SRC
        @cd common/crypto/arcfour; pwd; $(MAKE) EXPORT_SRC
        @cd common/crypto/blowfish;       pwd; $(MAKE) EXPORT_SRC
        @cd common/crypto/des;    pwd; $(MAKE) EXPORT_SRC
        @cd common/crypto/rsa;    pwd; $(MAKE) EXPORT_SRC
        @cd lib/crypt_modules/bsdbf; pwd ; $(MAKE) EXPORT_SRC
        @cd lib/gss_mechs/mech_dummy;   pwd; $(MAKE) EXPORT_SRC
        @cd lib/gss_mechs/mech_dh/backend;      pwd; $(MAKE) EXPORT_SRC
```

```
256         @cd lib/gss_mechs/mech_krb5;            pwd; $(MAKE) EXPORT_SRC
257         @cd lib/gss_mechs/mech_spnego;          pwd; $(MAKE) EXPORT_SRC
258         @cd lib/libcrypt; pwd; $(MAKE) EXPORT_SRC
259         @cd lib/libgss;   pwd; $(MAKE) EXPORT_SRC
260         @cd $(CLOSED)/lib/libike;   pwd; $(MAKE) EXPORT_SRC
261         @cd lib/libnsl;   pwd; $(MAKE) EXPORT_SRC
262         @cd lib/pkcs11/pkcs11_softtoken/common;  pwd; $(MAKE) EXPORT_SRC
263         @cd lib/libsldap; pwd; $(MAKE) EXPORT_SRC
264         @cd lib/libsasl; pwd; $(MAKE) EXPORT_SRC
265         @cd lib/sasl_plugins; pwd; $(MAKE) EXPORT_SRC
266         @cd lib/pam_modules/krb5;   pwd; $(MAKE) EXPORT_SRC
267         @cd psm/stand/boot; pwd; $(MAKE) EXPORT_SRC
268         @cd uts/common/crypto/io; pwd; $(MAKE) EXPORT_SRC
269         @cd uts/common/des; pwd; $(MAKE) EXPORT_SRC
270         @cd uts/common/rpc; pwd; $(MAKE) EXPORT_SRC
271         @cd uts/common/sys; pwd; $(MAKE) EXPORT_SRC
272         @cd uts/common/gssapi/include;          pwd; $(MAKE) EXPORT_SRC
273         @cd uts/common/gssapi;                  pwd; $(MAKE) EXPORT_SRC
274         @cd uts/common/gssapi/mechs/dummy;      pwd; $(MAKE) EXPORT_SRC
275         @cd uts/common/gssapi/mechs/krb5;       pwd; $(MAKE) EXPORT_SRC
276         @cd uts/common; pwd; $(MAKE) EXPORT_SRC
277         @cd uts/sparc; pwd; $(MAKE) EXPORT_SRC
278         @cd $(CLOSED)/uts/sun4u/forthdebug; pwd; $(MAKE) EXPORT_SRC
279         @cd $(CLOSED)/uts/sun4v/forthdebug; pwd; $(MAKE) EXPORT_SRC
280         @cd uts/intel; pwd; $(MAKE) EXPORT_SRC
281         @cd uts/sun4u; pwd; $(MAKE) EXPORT_SRC
282         @cd $(CLOSED)/uts/sun4v/io/ncp; pwd;    $(MAKE) EXPORT_SRC
283         @cd $(CLOSED)/uts/sun4v/io/n2cp; pwd;   $(MAKE) EXPORT_SRC
284         @cd pkg; pwd;   $(MAKE) EXPORT_SRC
285         $(RM) -r $(XMOD_DELETE_FILES)
286         $(RM) Targetdirs+
287         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
288                         < Targetdirs > Targetdirs+
289         $(MV) Targetdirs+ Targetdirs
290         $(CHMOD) 444 Targetdirs
291         $(RM) Makefile+
292         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
293                         < Makefile > Makefile+
294         $(MV) Makefile+ Makefile
295         $(CHMOD) 444 Makefile
296         $(RM) Makefile.master+
297         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
298                         < Makefile.master > Makefile.master+
299         $(MV) Makefile.master+ Makefile.master
300         $(CHMOD) 444 Makefile.master

302 CRYPT_SRC:
303         @cd $(CLOSED)/cmd/cmd-crypto/etc;  pwd; $(MAKE) CRYPT_SRC
304         @cd $(CLOSED)/cmd/cmd-inet/usr.lib/in.iked;  pwd; $(MAKE) CRYPT_SRC
305         @cd $(CLOSED)/cmd/cmd-inet/usr.lib/ike-certutils;  pwd; \
306             $(MAKE) CRYPT_SRC
307         @cd lib/crypt_modules/bsdbf; pwd ; $(MAKE) CRYPT_SRC
308         @cd lib/gss_mechs/mech_dummy;   pwd; $(MAKE) CRYPT_SRC
309         @cd lib/gss_mechs/mech_dh/backend; pwd; $(MAKE) CRYPT_SRC
310         @cd lib/gss_mechs/mech_krb5; pwd; $(MAKE) CRYPT_SRC
311         @cd lib/gss_mechs/mech_spnego; pwd; $(MAKE) CRYPT_SRC
312         @cd $(CLOSED)/lib/libike;   pwd; $(MAKE) CRYPT_SRC
313         @cd lib/libnsl;  pwd; $(MAKE) CRYPT_SRC
314         @cd lib/libsasl;        pwd; $(MAKE) CRYPT_SRC
315         @cd lib/sasl_plugins; pwd; $(MAKE) CRYPT_SRC
316         @cd lib/pam_modules/krb5;   pwd; $(MAKE) CRYPT_SRC
317         @cd uts/common/gssapi;    pwd; $(MAKE) CRYPT_SRC
318         @cd uts/common/gssapi/include;    pwd; $(MAKE) CRYPT_SRC
319         @cd uts/common/gssapi/mechs/dummy;        pwd; $(MAKE) CRYPT_SRC
320         @cd uts/common/gssapi/mechs/krb5; pwd; $(MAKE) CRYPT_SRC
321         $(RM) Makefile+
```



```
322         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
323                         < Makefile > Makefile+
324         $(MV) Makefile+ Makefile
325         $(CHMOD) 444 Makefile
326         $(RM) Makefile.master+
327         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
328                         < Makefile.master > Makefile.master+
329         $(MV) Makefile.master+ Makefile.master
330         $(CHMOD) 444 Makefile.master

332 # EXPORT DELETE END

334 ONC_PLUS:
335         @cd cmd/login; pwd; $(MAKE) ONC_PLUS
336         @cd uts; pwd; $(MAKE) ONC_PLUS

221 #
222 # Targets for reporting compiler versions; nightly uses these.
223 #

225 cc-version:
226         @if $($(MACH)_CC) -_versions >/dev/null 2>/dev/null; then \
227                 $(ECHO) 32-bit compiler;                        \
228                 $(ECHO) $($(MACH)_CC);                          \
229                 $($(MACH)_CC) -_versions 2>&1 |                 \
230                     $(EGREP) '^(cw|cc|gcc|primary|shadow)';     \
231         else                                                    \
232                 __COMPILER=`$($(MACH)_CC) -_compiler 2>/dev/null || $(TRUE)`;\
233                 if [ -z "$$__COMPILER" ]; then                  \
234                         $(ECHO) No 32-bit compiler found;       \
235                         exit 1;                                 \
236                 else                                            \
237                         $(ECHO) 32-bit compiler;                \
238                         $(ECHO) $($(MACH)_CC);                  \
239                         $(ECHO) $$__COMPILER;                   \
240                         $($(MACH)_CC) -V 2>&1 | head -1;        \
241                 fi;                                             \
242         fi

244 cc64-version:
245         @if $($(MACH64)_CC) -_versions >/dev/null 2>/dev/null; then \
246                 $(ECHO) 64-bit compiler;                        \
247                 $(ECHO) $($(MACH64)_CC);                        \
248                 $($(MACH64)_CC) -_versions 2>&1 |               \
249                     $(EGREP) '^(cw|cc|gcc|primary|shadow)';     \
250         else                                                    \
251                 __COMPILER=`$($(MACH64)_CC) -_compiler 2>/dev/null || $(TRUE)`;\
252                 if [ -z "$$__COMPILER" ]; then                  \
253                         $(ECHO) No 64-bit compiler found;       \
254                         exit 1;                                 \
255                 else                                            \
256                         $(ECHO) 64-bit compiler;                \
257                         $(ECHO) $($(MACH64)_CC);                \
258                         $(ECHO) $$__COMPILER;                   \
259                         $($(MACH64)_CC) -V 2>&1 | head -1;      \
260                 fi;                                             \
261         fi

263 java-version:
264         @if [ -x "$(JAVAC)" ]; then                     \
265                 $(ECHO) $(JAVAC);                       \
266                 $(JAVA_ROOT)/bin/java -fullversion 2>&1 | head -1;      \
267         else                                            \
268                 $(ECHO) No Java compiler found;         \
269                 exit 1;                                 \
270         fi
```

```
**********************************************************
   35775 Thu Jul 11 01:28:49 2013
new/usr/src/Makefile.master
onc plus-be-gone
**********************************************************
```

```
#
# 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.
#

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

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

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

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

134 JAVA_ROOT=      /usr/java

136 SFW_ROOT=       /usr/sfw
137 SFWINCDIR=      $(SFW_ROOT)/include
138 SFWLIBDIR=      $(SFW_ROOT)/lib
139 SFWLIBDIR64=    $(SFW_ROOT)/lib/$(MACH64)

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

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

147 RPCGEN=         /usr/bin/rpcgen
148 STABS=          $(ONBLD_TOOLS)/bin/$(MACH)/stabs
149 ELFEXTRACT=     $(ONBLD_TOOLS)/bin/$(MACH)/elfextract
150 MBH_PATCH=      $(ONBLD_TOOLS)/bin/$(MACH)/mbh_patch
151 ECHO=           echo
152 INS=            install
153 TRUE=           true
154 SYMLINK=        /usr/bin/ln -s
155 LN=             /usr/bin/ln
156 CHMOD=          /usr/bin/chmod
157 MV=             /usr/bin/mv -f
158 RM=             /usr/bin/rm -f
159 CUT=            /usr/bin/cut
160 NM=             /usr/ccs/bin/nm
161 DIFF=           /usr/bin/diff
162 GREP=           /usr/bin/grep
163 EGREP=          /usr/bin/egrep
164 ELFWRAP=        /usr/bin/elfwrap
165 KSH93=          /usr/bin/ksh93
166 SED=            /usr/bin/sed
167 NAWK=           /usr/bin/nawk
168 CP=             /usr/bin/cp -f
169 MCS=            /usr/ccs/bin/mcs
170 CAT=            /usr/bin/cat
171 ELFDUMP=        /usr/ccs/bin/elfdump
172 M4=             /usr/ccs/bin/m4
173 STRIP=          /usr/ccs/bin/strip
174 LEX=            /usr/ccs/bin/lex
175 FLEX=           $(SFW_ROOT)/bin/flex
176 YACC=           /usr/ccs/bin/yacc
177 CPP=            /usr/lib/cpp
178 JAVAC=          $(JAVA_ROOT)/bin/javac
179 JAVAH=          $(JAVA_ROOT)/bin/javah
180 JAVADOC=        $(JAVA_ROOT)/bin/javadoc
181 RMIC=           $(JAVA_ROOT)/bin/rmic
182 JAR=            $(JAVA_ROOT)/bin/jar
183 CTFCONVERT=     $(ONBLD_TOOLS)/bin/$(MACH)/ctfconvert
184 CTFMERGE=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfmerge
185 CTFSTABS=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfstabs
186 CTFSTRIP=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfstrip
187 NDRGEN=         $(ONBLD_TOOLS)/bin/$(MACH)/ndrgen
188 GENOFFSETS=     $(ONBLD_TOOLS)/bin/genoffsets
189 CTFCVTPTBL=     $(ONBLD_TOOLS)/bin/ctfcvtptbl
190 CTFFINDMOD=     $(ONBLD_TOOLS)/bin/ctffindmod
191 XREF=           $(ONBLD_TOOLS)/bin/xref
192 FIND=           /usr/bin/find
193 PERL=           /usr/bin/perl
```



```
194 PYTHON_26=      /usr/bin/python2.6
195 PYTHON=         $(PYTHON_26)
196 SORT=           /usr/bin/sort
197 TOUCH=          /usr/bin/touch
198 WC=             /usr/bin/wc
199 XARGS=          /usr/bin/xargs
200 ELFEDIT=        /usr/bin/elfedit
201 ELFSIGN=        /usr/bin/elfsign
202 DTRACE=         /usr/sbin/dtrace -xnolibs
203 UNIQ=           /usr/bin/uniq
204 TAR=            /usr/bin/tar

206 FILEMODE=       644
207 DIRMODE=        755

209 #
210 # The version of the patch makeup table optimized for build-time use.  Used
211 # during patch builds only.
212 $(PATCH_BUILD)PMTMO_FILE=$(SRC)/patch_makeup_table.mo

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

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

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

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

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

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

247 #
248 # IMPORTANT:: If you change any of INS.file, INS.dir, INS.rename,
249 # INS.link or INS.symlink here, then you must also change the
250 # corresponding override definitions in $CLOSED/Makefile.tonic.
251 # If you do not do this, then the closedbins build for the OpenSolaris
252 # community will break. PS, the gatekeepers will be upset too.
253 INS.file=       $(RM) $@; $(INS) -s -m $(FILEMODE) -f $(@D) $<
254 INS.dir=        $(INS) -s -d -m $(DIRMODE) $@
255 # installs and renames at once
256 #
257 INS.rename=     $(INS.file); $(MV) $(@D)/$(<F) $@

259 # install a link
```

```
260 INSLINKTARGET=  $<
261 INS.link=       $(RM) $@; $(LN) $(INSLINKTARGET) $@
262 INS.symlink=    $(RM) $@; $(SYMLINK) $(INSLINKTARGET) $@

264 #
265 # Python bakes the mtime of the .py file into the compiled .pyc and
266 # rebuilds if the baked-in mtime != the mtime of the source file
267 # (rather than only if it's less than), thus when installing python
268 # files we must make certain to not adjust the mtime of the source
269 # (.py) file.
270 #
271 INS.pyfile=     $(INS.file); $(TOUCH) -r $< $@

273 # MACH must be set in the shell environment per uname -p on the build host
274 # More specific architecture variables should be set in lower makefiles.
275 #
276 # MACH64 is derived from MACH, and BUILD64 is set to '#' for
277 # architectures on which we do not build 64-bit versions.
278 # (There are no such architectures at the moment.)
279 #
280 # Set BUILD64=# in the environment to disable 64-bit amd64
281 # builds on i386 machines.

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

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

289 sparc_BUILD64=
290 i386_BUILD64=
291 BUILD64=        $($(MACH)_BUILD64)

293 #
294 # C compiler mode. Future compilers may change the default on us,
295 # so force extended ANSI mode globally. Lower level makefiles can
296 # override this by setting CCMODE.
297 #
298 CCMODE=                 -Xa
299 CCMODE64=               -Xa

301 #
302 # C compiler verbose mode. This is so we can enable it globally,
303 # but turn it off in the lower level makefiles of things we cannot
304 # (or aren't going to) fix.
305 #
306 CCVERBOSE=              -v

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

312 # set this to the secret flag "-Wc,-Qiselect-regsym=0" to disable register
313 # symbols (used to detect conflicts between objects that use global registers)
314 # we disable this now for safety, and because genunix doesn't link with
315 # this feature (the v9 default) enabled.
316 #
317 # REGSYM is separate since the C++ driver syntax is different.
318 CCREGSYM=               -Wc,-Qiselect-regsym=0
319 CCCREGSYM=              -Qoption cg -Qiselect-regsym=0

321 # Prevent the removal of static symbols by the SPARC code generator (cg).
322 # The x86 code generator (ube) does not remove such symbols and as such
323 # using this workaround is not applicable for x86.
324 #
325 CCSTATICSYM=            -Wc,-Qassembler-ounrefsym=0
```



```
326 #
327 # generate 32-bit addresses in the v9 kernel. Saves memory.
328 CCABS32=                -Wc,-xcode=abs32
329 #
330 # generate v9 code which tolerates callers using the v7 ABI, for the sake of
331 # system calls.
332 CC32BITCALLERS=         -_gcc=-massume-32bit-callers

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

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

358 #
359 # compiler '-xarch' flag. This is here to centralize it and make it
360 # overridable for testing.
361 sparc_XARCH=    -m32
362 sparcv9_XARCH=  -m64
363 i386_XARCH=
364 amd64_XARCH=    -m64 -Ui386 -U__i386

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

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

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

390 STAND_FLAGS_32 = $($(MACH)_STAND_FLAGS)
391 STAND_FLAGS_64 = $($(MACH64)_STAND_FLAGS)
```

```
393 #
394 # disable the incremental linker
395 ILDOFF=                 -xildoff
396 #
397 XDEPEND=                -xdepend
398 XFFLAG=                 -xF=%all
399 XESS=                   -xs
400 XSTRCONST=              -xstrconst

402 #
403 # turn warnings into errors (C)
404 CERRWARN = -errtags=yes -errwarn=%all
405 CERRWARN += -erroff=E_EMPTY_TRANSLATION_UNIT
406 CERRWARN += -erroff=E_STATEMENT_NOT_REACHED

408 CERRWARN += -_gcc=-Wno-missing-braces
409 CERRWARN += -_gcc=-Wno-sign-compare
410 CERRWARN += -_gcc=-Wno-unknown-pragmas
411 CERRWARN += -_gcc=-Wno-unused-parameter
412 CERRWARN += -_gcc=-Wno-missing-field-initializers

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

417 # DEBUG v. -nd make for frequent unused variables, empty conditions, etc. in
418 # -nd builds
419 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-unused
420 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-empty-body

422 #
423 # turn warnings into errors (C++)
424 CCERRWARN=              -xwe

426 # C99 mode
427 C99_ENABLE=     -xc99=%all
428 C99_DISABLE=    -xc99=%none
429 C99MODE=        $(C99_DISABLE)
430 C99LMODE=       $(C99MODE:-xc99%=-Xc99%)

432 # In most places, assignments to these macros should be appended with +=
433 # (CPPFLAGS.master allows values to be prepended to CPPFLAGS).
434 sparc_CFLAGS=   $(sparc_XARCH) $(CCSTATICSYM)
435 sparcv9_CFLAGS= $(sparcv9_XARCH) -dalign $(CCVERBOSE) $(V9ABIWARN) $(CCREGSYM) \
436                 $(CCSTATICSYM)
437 i386_CFLAGS=    $(i386_XARCH)
438 amd64_CFLAGS=   $(amd64_XARCH)

440 sparc_ASFLAGS=  $(sparc_AS_XARCH)
441 sparcv9_ASFLAGS=$(sparcv9_AS_XARCH)
442 i386_ASFLAGS=   $(i386_AS_XARCH)
443 amd64_ASFLAGS=  $(amd64_AS_XARCH)

445 #
446 sparc_COPTFLAG=         -xO3
447 sparcv9_COPTFLAG=       -xO3
448 i386_COPTFLAG=          -O
449 amd64_COPTFLAG=         -xO3

451 COPTFLAG= $($(MACH)_COPTFLAG)
452 COPTFLAG64= $($(MACH64)_COPTFLAG)

454 # When -g is used, the compiler globalizes static objects
455 # (gives them a unique prefix). Disable that.
456 CNOGLOBAL= -W0,-noglobal
```



```
458 # Direct the Sun Studio compiler to use a static globalization prefix based on t
459 # name of the module rather than something unique. Otherwise, objects
460 # will not build deterministically, as subsequent compilations of identical
461 # source will yeild objects that always look different.
462 #
463 # In the same spirit, this will also remove the date from the N_OPT stab.
464 CGLOBALSTATIC= -W0,-xglobalstatic

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

470 #
471 # Default debug format for Sun Studio 11 is dwarf, so force it to
472 # generate stabs.
473 #
474 DEBUGFORMAT=    -xdebugformat=stabs

476 #
477 # Flags used to build in debug mode for ctf generation.  Bugs in the Devpro
478 # compilers currently prevent us from building with cc-emitted DWARF.
479 #
480 CTF_FLAGS_sparc = -g -Wc,-Qiselect-T1 $(C99MODE) $(CNOGLOBAL) $(CDWARFSTR)
481 CTF_FLAGS_i386  = -g $(C99MODE) $(CNOGLOBAL) $(CDWARFSTR)

483 CTF_FLAGS_sparcv9       = $(CTF_FLAGS_sparc)
484 CTF_FLAGS_amd64         = $(CTF_FLAGS_i386)

486 # Sun Studio produces broken userland code when saving arguments.
487 $(__GNUC)CTF_FLAGS_amd64 += $(SAVEARGS)

489 CTF_FLAGS_32    = $(CTF_FLAGS_$(MACH)) $(DEBUGFORMAT)
490 CTF_FLAGS_64    = $(CTF_FLAGS_$(MACH64)) $(DEBUGFORMAT)
491 CTF_FLAGS       = $(CTF_FLAGS_32)

493 #
494 # Flags used with genoffsets
495 #
496 GOFLAGS = -_noecho \
497         $(CALLSYMS) \
498         $(CDWARFSTR)

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

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

506 #
507 # tradeoff time for space (smaller is better)
508 #
509 sparc_SPACEFLAG         = -xspace -W0,-Lt
510 sparcv9_SPACEFLAG       = -xspace -W0,-Lt
511 i386_SPACEFLAG          = -xspace
512 amd64_SPACEFLAG         =

514 SPACEFLAG               = $($(MACH)_SPACEFLAG)
515 SPACEFLAG64             = $($(MACH64)_SPACEFLAG)

517 #
518 # The Sun Studio 11 compiler has changed the behaviour of integer
519 # wrap arounds and so a flag is needed to use the legacy behaviour
520 # (without this flag panics/hangs could be exposed within the source).
521 #
522 sparc_IROPTFLAG         = -W2,-xwrap_int
523 sparcv9_IROPTFLAG       = -W2,-xwrap_int
```

```
524 i386_IROPTFLAG          =
525 amd64_IROPTFLAG         =

527 IROPTFLAG               = $($(MACH)_IROPTFLAG)
528 IROPTFLAG64             = $($(MACH64)_IROPTFLAG)

530 sparc_XREGSFLAG         = -xregs=no%appl
531 sparcv9_XREGSFLAG       = -xregs=no%appl
532 i386_XREGSFLAG          =
533 amd64_XREGSFLAG         =

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

538 CFLAGS=         $(COPTFLAG) $($(MACH)_CFLAGS) $(SPACEFLAG) $(CCMODE) \
539                 $(ILDOFF) $(CERRWARN) $(C99MODE) $(CCUNBOUND) $(IROPTFLAG) \
540                 $(CGLOBALSTATIC) $(CCNOAUTOINLINE)
541 CFLAGS64=       $(COPTFLAG64) $($(MACH64)_CFLAGS) $(SPACEFLAG64) $(CCMODE64) \
542                 $(ILDOFF) $(CERRWARN) $(C99MODE) $(CCUNBOUND) $(IROPTFLAG64) \
543                 $(CGLOBALSTATIC) $(CCNOAUTOINLINE)
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)

552 DTEXTDOM=-DTEXT_DOMAIN=\"$(TEXT_DOMAIN)\"       # For messaging.
553 DTS_ERRNO=-D_TS_ERRNO
554 CPPFLAGS.master=$(DTEXTDOM) $(DTS_ERRNO) \
555         $(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) $(ENVCPPFLAGS4)
556 CPPFLAGS.native=$(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) $(ENVCPPFLAGS4)
557 CPPFLAGS=       $(CPPFLAGS.master)
558 AS_CPPFLAGS=    $(CPPFLAGS.master)
559 JAVAFLAGS=      -deprecation

561 #
562 # For source message catalogue
563 #
564 .SUFFIXES: $(SUFFIXES) .i .po
565 MSGROOT= $(ROOT)/catalog
566 MSGDOMAIN= $(MSGROOT)/$(TEXT_DOMAIN)
567 MSGDOMAINPOFILE = $(MSGDOMAIN)/$(POFILE)
568 DCMSGDOMAIN= $(MSGROOT)/LC_TIME/$(TEXT_DOMAIN)
569 DCMSGDOMAINPOFILE = $(DCMSGDOMAIN)/$(DCFILE:.dc=.po)

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

583 #
584 # This is overwritten by local Makefile when PROG is a list.
585 #
586 POFILE= $(PROG).po

588 sparc_CCFLAGS=          -cg92 -compat=4 \
589                         -Qoption ccfe -messages=no%anachronism \
```



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

605 sparc_CCOPTFLAG=        -O
606 sparcv9_CCOPTFLAG=      -O
607 i386_CCOPTFLAG=         -O
608 amd64_CCOPTFLAG=        -O

610 CCOPTFLAG=      $($(MACH)_CCOPTFLAG)
611 CCOPTFLAG64=    $($(MACH64)_CCOPTFLAG)
612 CCFLAGS=        $(CCOPTFLAG) $($(MACH)_CCFLAGS)
613 CCFLAGS64=      $(CCOPTFLAG64) $($(MACH64)_CCFLAGS)

615 #
616 #
617 #
618 ELFWRAP_FLAGS   =
619 ELFWRAP_FLAGS64 =       -64

621 #
622 # Various mapfiles that are used throughout the build, and delivered to
623 # /usr/lib/ld.
624 #
625 MAPFILE.NED_i386 =      $(SRC)/common/mapfiles/common/map.noexdata
626 MAPFILE.NED_sparc =
627 MAPFILE.NED =           $(MAPFILE.NED_$(MACH))
628 MAPFILE.PGA =           $(SRC)/common/mapfiles/common/map.pagealign
629 MAPFILE.NES =           $(SRC)/common/mapfiles/common/map.noexstk
630 MAPFILE.FLT =           $(SRC)/common/mapfiles/common/map.filter
631 MAPFILE.LEX =           $(SRC)/common/mapfiles/common/map.lex.yy

633 #
634 # Generated mapfiles that are compiler specific, and used throughout the
635 # build.  These mapfiles are not delivered in /usr/lib/ld.
636 #
637 MAPFILE.NGB_sparc=      $(SRC)/common/mapfiles/gen/sparc_cc_map.noexeglobs
638 $(__GNUC64)MAPFILE.NGB_sparc= \
639                         $(SRC)/common/mapfiles/gen/sparc_gcc_map.noexeglobs
640 MAPFILE.NGB_sparcv9=    $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexeglobs
641 $(__GNUC64)MAPFILE.NGB_sparcv9= \
642                         $(SRC)/common/mapfiles/gen/sparcv9_gcc_map.noexeglobs
643 MAPFILE.NGB_i386=       $(SRC)/common/mapfiles/gen/i386_cc_map.noexeglobs
644 $(__GNUC64)MAPFILE.NGB_i386= \
645                         $(SRC)/common/mapfiles/gen/i386_gcc_map.noexeglobs
646 MAPFILE.NGB_amd64=      $(SRC)/common/mapfiles/gen/amd64_cc_map.noexeglobs
647 $(__GNUC64)MAPFILE.NGB_amd64= \
648                         $(SRC)/common/mapfiles/gen/amd64_gcc_map.noexeglobs
649 MAPFILE.NGB =           $(MAPFILE.NGB_$(MACH))

651 #
652 # A generic interface mapfile name, used by various dynamic objects to define
653 # the interfaces and interposers the object must export.
654 #
655 MAPFILE.INT =           mapfile-intf
```

```
657 #
658 # LDLIBS32 can be set in the environment to override the following assignment.
659 # LDLIBS64 can be set to override the assignment made in Makefile.master.64.
660 # These environment settings make sure that no libraries are searched outside
661 # of the local workspace proto area:
662 #       LDLIBS32=-YP,$ROOT/lib:$ROOT/usr/lib
663 #       LDLIBS64=-YP,$ROOT/lib/$MACH64:$ROOT/usr/lib/$MACH64
664 #
665 LDLIBS32 =      $(ENVLDLIBS1) $(ENVLDLIBS2) $(ENVLDLIBS3)
666 LDLIBS.cmd =    $(LDLIBS32)
667 LDLIBS.lib =    $(LDLIBS32)
668 #
669 # Define compilation macros.
670 #
671 COMPILE.c=      $(CC) $(CFLAGS) $(CPPFLAGS) -c
672 COMPILE64.c=    $(CC) $(CFLAGS64) $(CPPFLAGS) -c
673 COMPILE.cc=     $(CCC) $(CCFLAGS) $(CPPFLAGS) -c
674 COMPILE64.cc=   $(CCC) $(CCFLAGS64) $(CPPFLAGS) -c
675 COMPILE.s=      $(AS) $(ASFLAGS) $(AS_CPPFLAGS)
676 COMPILE64.s=    $(AS) $(ASFLAGS) $($(MACH64)_AS_XARCH) $(AS_CPPFLAGS)
677 COMPILE.d=      $(DTRACE) -G -32
678 COMPILE64.d=    $(DTRACE) -G -64
679 COMPILE.b=      $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))
680 COMPILE64.b=    $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))

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

685 #
686 # Link time macros
687 #
688 CCNEEDED                = -lC
689 CCEXTNEEDED             = -lCrun -lCstd
690 $(__GNUC)CCNEEDED       = -L$(GCCLIBDIR) -R$(GCCLIBDIR) -lstdc++ -lgcc_s
691 $(__GNUC)CCEXTNEEDED    = $(CCNEEDED)

693 LINK.c=         $(CC) $(CFLAGS) $(CPPFLAGS) $(LDFLAGS)
694 LINK64.c=       $(CC) $(CFLAGS64) $(CPPFLAGS) $(LDFLAGS)
695 NORUNPATH=      -norunpath -nolib
696 LINK.cc=        $(CCC) $(CCFLAGS) $(CPPFLAGS) $(NORUNPATH) \
697                 $(LDFLAGS) $(CCNEEDED)
698 LINK64.cc=      $(CCC) $(CCFLAGS64) $(CPPFLAGS) $(NORUNPATH) \
699                 $(LDFLAGS) $(CCNEEDED)

701 #
702 # lint macros
703 #
704 # Note that the undefine of __PRAGMA_REDEFINE_EXTNAME can be removed once
705 # ON is built with a version of lint that has the fix for 4484186.
706 #
707 ALWAYS_LINT_DEFS =      -errtags=yes -s
708 ALWAYS_LINT_DEFS +=     -erroff=E_PTRDIFF_OVERFLOW
709 ALWAYS_LINT_DEFS +=     -erroff=E_ASSIGN_NARROW_CONV
710 ALWAYS_LINT_DEFS +=     -U__PRAGMA_REDEFINE_EXTNAME
711 ALWAYS_LINT_DEFS +=     $(C99LMODE)
712 ALWAYS_LINT_DEFS +=     -errsecurity=$(SECLEVEL)
713 ALWAYS_LINT_DEFS +=     -erroff=E_SEC_CREAT_WITHOUT_EXCL
714 ALWAYS_LINT_DEFS +=     -erroff=E_SEC_FORBIDDEN_WARN_CREAT
715 # XX64 -- really only needed for amd64 lint
716 ALWAYS_LINT_DEFS +=     -erroff=E_ASSIGN_INT_TO_SMALL_INT
717 ALWAYS_LINT_DEFS +=     -erroff=E_CAST_INT_CONST_TO_SMALL_INT
718 ALWAYS_LINT_DEFS +=     -erroff=E_CAST_INT_TO_SMALL_INT
719 ALWAYS_LINT_DEFS +=     -erroff=E_CAST_TO_PTR_FROM_INT
720 ALWAYS_LINT_DEFS +=     -erroff=E_COMP_INT_WITH_LARGE_INT
721 ALWAYS_LINT_DEFS +=     -erroff=E_INTEGRAL_CONST_EXP_EXPECTED
```



```
722 ALWAYS_LINT_DEFS +=     -erroff=E_PASS_INT_TO_SMALL_INT
723 ALWAYS_LINT_DEFS +=     -erroff=E_PTR_CONV_LOSES_BITS

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

729 SECLEVEL=       core
730 LINT.c=         $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS) $(CPPFLAGS) \
731                 $(ALWAYS_LINT_DEFS)
732 LINT64.c=       $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS64) $(CPPFLAGS) \
733                 $(ALWAYS_LINT_DEFS)
734 LINT.s=         $(LINT.c)

736 # For some future builds, NATIVE_MACH and MACH might be different.
737 # Therefore, NATIVE_MACH needs to be redefined in the
738 # environment as ‘uname -p‘ to override this macro.
739 #
740 # For now at least, we cross-compile amd64 on i386 machines.
741 NATIVE_MACH=    $(MACH:amd64=i386)

743 # Define native compilation macros
744 #

746 # Base directory where compilers are loaded.
747 # Defined here so it can be overridden by developer.
748 #
749 SPRO_ROOT=              $(BUILD_TOOLS)/SUNWspro
750 SPRO_VROOT=             $(SPRO_ROOT)/SS12
751 GNU_ROOT=               $(SFW_ROOT)

753 # Till SS12u1 formally becomes the NV CBE, LINT is hard
754 # coded to be picked up from the $SPRO_ROOT/sunstudio12.1/
755 # location. Impacted variables are sparc_LINT, sparcv9_LINT,
756 # i386_LINT, amd64_LINT.
757 # Reset them when SS12u1 is rolled out.
758 #

760 # Specify platform compiler versions for languages
761 # that we use (currently only c and c++).
762 #
763 sparc_CC=               $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
764 $(__GNUC)sparc_CC=      $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
765 sparc_CCC=              $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
766 $(__GNUC)sparc_CCC=     $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
767 sparc_CPP=              /usr/ccs/lib/cpp
768 sparc_AS=               /usr/ccs/bin/as -xregsym=no
769 sparc_LD=               /usr/ccs/bin/ld
770 sparc_LINT=             $(SPRO_ROOT)/sunstudio12.1/bin/lint

772 sparcv9_CC=             $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
773 $(__GNUC64)sparcv9_CC=  $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
774 sparcv9_CCC=            $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
775 $(__GNUC64)sparcv9_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
776 sparcv9_CPP=            /usr/ccs/lib/cpp
777 sparcv9_AS=             /usr/ccs/bin/as -xregsym=no
778 sparcv9_LD=             /usr/ccs/bin/ld
779 sparcv9_LINT=           $(SPRO_ROOT)/sunstudio12.1/bin/lint

781 i386_CC=               $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
782 $(__GNUC)i386_CC=      $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
783 i386_CCC=              $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
784 $(__GNUC)i386_CCC=     $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
785 i386_CPP=              /usr/ccs/lib/cpp
786 i386_AS=               /usr/ccs/bin/as
787 $(__GNUC)i386_AS=      $(ONBLD_TOOLS)/bin/$(MACH)/aw
```

```
i386_LD=                /usr/ccs/bin/ld
i386_LINT=              $(SPRO_ROOT)/sunstudio12.1/bin/lint

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

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

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

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

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

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

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



```
GSHARED=        -G
CCMT=           -mt

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

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

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

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

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

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

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

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

ELFSIGN_O=      $(TRUE)
ELFSIGN_CRYPTO= $(ELFSIGN_O)
ELFSIGN_OBJECT= $(ELFSIGN_O)
$(EXPORT_RELEASE_BUILD)ELFSIGN_O =      $(ELFSIGN)
$(EXPORT_RELEASE_BUILD)ELFSIGN_CFNAME = SUNWosnetCF
$(EXPORT_RELEASE_BUILD)ELFSIGN_KEY =    \
                        $(CLOSED)/cmd/cmd-crypto/etc/keys/$(ELFSIGN_CFNAME)
$(EXPORT_RELEASE_BUILD)ELFSIGN_CERT=    \
                        $(CLOSED)/cmd/cmd-crypto/etc/certs/$(ELFSIGN_CFNAME)
$(EXPORT_RELEASE_BUILD)ELFSIGN_SENAME = SUNWosnetSE
$(EXPORT_RELEASE_BUILD)ELFSIGN_SEKEY =  \
                        $(CLOSED)/cmd/cmd-crypto/etc/keys/$(ELFSIGN_SENAME)
$(EXPORT_RELEASE_BUILD)ELFSIGN_SECERT=  \
                        $(CLOSED)/cmd/cmd-crypto/etc/certs/$(ELFSIGN_SENAME)
$(EXPORT_RELEASE_BUILD)ELFSIGN_CRYPTO=  $(ELFSIGN_O) sign \
                        $(ELFSIGN_FORMAT_OPTION) \
                        -k $(ELFSIGN_KEY) -c $(ELFSIGN_CERT) -e $@
$(EXPORT_RELEASE_BUILD)ELFSIGN_OBJECT=  $(ELFSIGN_O) sign \
                        $(ELFSIGN_FORMAT_OPTION) \
                        -k $(ELFSIGN_SEKEY) -c $(ELFSIGN_SECERT) -e $@
```

```
# 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} -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
```

```
1052 .l:
1053         $(RM) $*.c
1054         $(LEX.l) $< > $*.c
1055         $(LINK.c) -o $@ $*.c -ll $(LDLIBS)
1056         $(POST_PROCESS)
1057         $(RM) $*.c
1058 .l.o:
1059         $(RM) $*.c
1060         $(LEX.l) $< > $*.c
1061         $(COMPILE.c) -o $@ $*.c $(CTFCONVERT_HOOK)
1062         $(POST_PROCESS_O)
1063         $(RM) $*.c

1065 .bin.o:
1066         $(COMPILE.b) -o $@ $<
1067         $(POST_PROCESS_O)

1069 .java.class:
1070         $(COMPILE.java) $<

1072 # Bourne and Korn shell script message catalog build rules.
1073 # We extract all gettext strings with sed(1) (being careful to permit
1074 # multiple gettext strings on the same line), weed out the dups, and
1075 # build the catalogue with awk(1).

1077 .sh.po .ksh.po:
1078         $(SED) -n -e ":a"                                       \
1079                   -e "h"                                        \
1080                   -e "s/.*gettext *\(\"[^\"]*\"\).*/\1/p"       \
1081                   -e "x"                                        \
1082                   -e "s/\(.*\)gettext *\"[^\"]*\"\(.*\)/\1\2/"  \
1083                   -e "t a"                                      \
1084                $< | sort -u | awk '{ print "msgid\t" $$0 "\nmsgstr" }' > $@

1086 #
1087 # Python and Perl executable and message catalog build rules.
1088 #
1089 .SUFFIXES: .pl .pm .py .pyc

1091 .pl:
1092         $(RM) $@;
1093         $(SED) -e "s@TEXT_DOMAIN@\"$(TEXT_DOMAIN)\"@" $< > $@;
1094         $(CHMOD) +x $@

1096 .py:
1097         $(RM) $@; $(CAT) $< > $@; $(CHMOD) +x $@

1099 .py.pyc:
1100         $(RM) $@
1101         $(PYTHON) -mpy_compile $<
1102         @[ $(<)c = $@ ] || $(MV) $(<)c $@

1104 .py.po:
1105         $(GNUXGETTEXT) $(GNUXGETFLAGS) -d $(<F:%.py=%) $< ;

1107 .pl.po .pm.po:
1108         $(XGETTEXT) $(XGETFLAGS) -d $(<F) $< ;
1109         $(RM)   $@ ;
1110         $(SED) "/^domain/d" < $(<F).po > $@ ;
1111         $(RM) $(<F).po

1113 #
1114 # When using xgettext, we want messages to go to the default domain,
1115 # rather than the specified one.  This special version of the
1116 # COMPILE.cpp macro effectively prevents expansion of TEXT_DOMAIN,
1117 # causing xgettext to put all messages into the default domain.
```



```
1118 #
1119 CPPFORPO=$(COMPILE.cpp:\"$(TEXT_DOMAIN)\"=TEXT_DOMAIN)

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

1124 .h.i:
1125         $(CPPFORPO) $< > $@

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

1132 .l.i:
1133         $(LEX) $<
1134         $(CPPFORPO) lex.yy.c  > $@
1135         $(RM) lex.yy.c

1137 .c.po:
1138         $(CPPFORPO) $< > $<.i
1139         $(BUILD.po)

1141 .y.po:
1142         $(YACC) -d $<
1143         $(CPPFORPO) y.tab.c  > $<.i
1144         $(BUILD.po)
1145         $(RM) y.tab.c

1147 .l.po:
1148         $(LEX) $<
1149         $(CPPFORPO) lex.yy.c  > $<.i
1150         $(BUILD.po)
1151         $(RM) lex.yy.c

1153 #
1154 # Rules to perform stylistic checks
1155 #
1156 .SUFFIXES: .x .xml .check .xmlchk

1158 .h.check:
1159         $(DOT_H_CHECK)

1161 .x.check:
1162         $(DOT_X_CHECK)

1164 .xml.xmlchk:
1165         $(MANIFEST_CHECK)

1167 #
1168 # Rules to process ONC+ Source partial files
1169 #
1170 %_onc_plus:     %
1171         @$(ECHO) "extracting code from $< ... "
1172         sed -n -e '/ONC_PLUS EXTRACT START/,/ONC_PLUS EXTRACT END/p' $<  > $@

1174 #
1168 # Include rules to render automated sccs get rules "safe".
1169 #
1170 include $(SRC)/Makefile.noget
```

```
**********************************************************
   70024 Thu Jul 11 01:28:49 2013
new/usr/src/Targetdirs
first pass
**********************************************************
```

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

  21 #
  22 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
  23 # Copyright 2011, Richard Lowe
  24 # Copyright 2011 Nexenta Systems, Inc. All rights reserved.
  25 # Copyright (c) 2012 by Delphix. All rights reserved.
  26 # Copyright 2012 OmniTI Computer Consulting, Inc.  All rights reserved.
  27 #

  29 #
  30 # It is easier to think in terms of directory names without the ROOT macro
  31 # prefix.  ROOTDIRS is TARGETDIRS with ROOT prefixes.  It is necessary
  32 # to work with ROOT prefixes when controlling conditional assignments.
  33 #

  35 DIRLINKS=       $(SYM.DIRS)
  36 $(BUILD64)      DIRLINKS += $(SYM.DIRS64)

  38 FILELINKS= $(SYM.USRCCSLIB) $(SYM.USRLIB)
  39 $(BUILD64)      FILELINKS += $(SYM.USRCCSLIB64) $(SYM.USRLIB64)

  41 TARGETDIRS=     $(DIRS)
  42 $(BUILD64)      TARGETDIRS += $(DIRS64)

  44 TARGETDIRS      += $(FILELINKS) $(DIRLINKS)

  46 i386_DIRS=                      \
  47         /boot/acpi              \
  48         /boot/acpi/tables       \
  49         /boot/grub              \
  50         /boot/grub/bin          \
  51         /platform/i86pc         \
  52         /usr/lib/xen            \
  53         /usr/lib/xen/bin

  55 sparc_DIRS=                            \
  56         /usr/lib/ldoms

  58 # EXPORT DELETE START
  59 XDIRS= \
  60         /usr/lib/inet/wanboot
  61 # EXPORT DELETE END
```



```
  58 sparc_64ONLY= $(POUND_SIGN)
  59 64ONLY=  $($(MACH)_64ONLY)

  61 $(64ONLY) MACH32_DIRS=/usr/ucb/$(MACH32)

  63 DIRS= \
  64         /boot \
  65         /boot/solaris \
  66         /boot/solaris/bin \
  67         $($(MACH)_DIRS) \
  68         /dev \
  69         /dev/dsk \
  70         /dev/fd \
  71         /dev/ipnet \
  72         /dev/net \
  73         /dev/rdsk \
  74         /dev/rmt \
  75         /dev/pts \
  76         /dev/sad \
  77         /dev/swap \
  78         /dev/term \
  79         /dev/vt \
  80         /dev/zcons \
  81         /devices \
  82         /devices/pseudo \
  83         /etc \
  84         /etc/brand  \
  85         /etc/brand/solaris10  \
  86         /etc/cron.d \
  87         /etc/crypto \
  88         /etc/crypto/certs \
  89         /etc/crypto/crls \
  90         /etc/dbus-1 \
  91         /etc/dbus-1/system.d \
  92         /etc/default \
  93         /etc/devices  \
  94         /etc/dev  \
  95         /etc/dfs  \
  96         /etc/dladm \
  97         /etc/fs  \
  98         /etc/fs/nfs  \
  99         /etc/fs/zfs \
 100         /etc/ftpd  \
 101         /etc/hal \
 102         /etc/hal/fdi \
 103         /etc/hal/fdi/information \
 104         /etc/hal/fdi/information/10freedesktop \
 105         /etc/hal/fdi/information/20thirdparty \
 106         /etc/hal/fdi/information/30user \
 107         /etc/hal/fdi/policy \
 108         /etc/hal/fdi/policy/10osvendor \
 109         /etc/hal/fdi/policy/20thirdparty \
 110         /etc/hal/fdi/policy/30user \
 111         /etc/hal/fdi/preprobe \
 112         /etc/hal/fdi/preprobe/10osvendor \
 113         /etc/hal/fdi/preprobe/20thirdparty \
 114         /etc/hal/fdi/preprobe/30user \
 115         /etc/ipadm \
 116         /etc/iscsi \
 117         /etc/rpcsec     \
 118         /etc/security   \
 119         /etc/security/auth_attr.d \
 120         /etc/security/exec_attr.d \
 121         /etc/security/prof_attr.d \
 122         /etc/security/tsol      \
```

```
123         /etc/gss        \
124         /etc/init.d  \
125         /etc/dhcp       \
126         /etc/lib  \
127         /etc/mail  \
128         /etc/mail/cf  \
129         /etc/mail/cf/cf  \
130         /etc/mail/cf/domain  \
131         /etc/mail/cf/feature  \
132         /etc/mail/cf/m4  \
133         /etc/mail/cf/mailer  \
134         /etc/mail/cf/ostype  \
135         /etc/mail/cf/sh  \
136         /etc/net-snmp \
137         /etc/net-snmp/snmp \
138         /etc/opt  \
139         /etc/rc0.d  \
140         /etc/rc1.d  \
141         /etc/rc2.d  \
142         /etc/rc3.d  \
143         /etc/rcS.d  \
144         /etc/saf \
145         /etc/sasl       \
146         /etc/sfw \
147         /etc/svc \
148         /etc/svc/profile \
149         /etc/svc/profile/site \
150         /etc/svc/volatile \
151         /etc/tm  \
152         /etc/usb   \
153         /etc/user_attr.d \
154         /etc/zfs  \
155         /etc/zones  \
156         /export  \
157         /home  \
158         /lib \
159         /lib/crypto \
160         /lib/inet \
161         /lib/fm \
162         /lib/secure \
163         /lib/svc \
164         /lib/svc/bin \
165         /lib/svc/capture \
166         /lib/svc/manifest \
167         /lib/svc/manifest/milestone \
168         /lib/svc/manifest/device \
169         /lib/svc/manifest/system \
170         /lib/svc/manifest/system/device \
171         /lib/svc/manifest/system/filesystem \
172         /lib/svc/manifest/system/security \
173         /lib/svc/manifest/system/svc \
174         /lib/svc/manifest/network \
175         /lib/svc/manifest/network/dns \
176         /lib/svc/manifest/network/ipsec \
177         /lib/svc/manifest/network/ldap \
178         /lib/svc/manifest/network/nfs \
179         /lib/svc/manifest/network/nis \
180         /lib/svc/manifest/network/rpc \
181         /lib/svc/manifest/network/security \
182         /lib/svc/manifest/network/shares \
183         /lib/svc/manifest/network/ssl \
184         /lib/svc/manifest/application \
185         /lib/svc/manifest/application/management \
186         /lib/svc/manifest/application/security \
187         /lib/svc/manifest/application/print \
188         /lib/svc/manifest/platform \
```



```
189         /lib/svc/manifest/platform/sun4u \
190         /lib/svc/manifest/platform/sun4v \
191         /lib/svc/manifest/site \
192         /lib/svc/method \
193         /lib/svc/monitor \
194         /lib/svc/seed \
195         /lib/svc/share \
196         /kernel  \
197         /mnt  \
198         /opt  \
199         /platform  \
200         /proc  \
201         /root  \
202         /sbin  \
203         /system \
204         /system/contract \
205         /system/object \
206         /tmp \
207         /usr  \
208         /usr/4lib \
209         /usr/ast \
210         /usr/ast/bin \
211         /usr/bin \
212         /usr/bin/$(MACH32) \
213         /usr/ccs \
214         /usr/ccs/bin \
215         /usr/ccs/lib \
216         /usr/demo \
217         /usr/demo/SOUND \
218         /usr/games \
219         /usr/has \
220         /usr/has/bin \
221         /usr/has/lib \
222         /usr/has/man \
223         /usr/has/man/man1has \
224         /usr/include \
225         /usr/include/ast \
226         /usr/include/fm \
227         /usr/include/gssapi \
228         /usr/include/hal \
229         /usr/include/kerberosv5 \
230         /usr/include/libmilter \
231         /usr/include/libpolkit \
232         /usr/include/sasl \
233         /usr/include/scsi \
234         /usr/include/security \
235         /usr/include/sys/crypto \
236         /usr/include/tsol \
237         /usr/kernel  \
238         /usr/kvm \
239         /usr/lib \
240         /usr/lib/abi \
241         /usr/lib/brand \
242         /usr/lib/brand/ipkg \
243         /usr/lib/brand/labeled \
244         /usr/lib/brand/shared \
245         /usr/lib/brand/sn1 \
246         /usr/lib/brand/solaris10 \
247         /usr/lib/class \
248         /usr/lib/class/FSS \
249         /usr/lib/class/FX \
250         /usr/lib/class/IA \
251         /usr/lib/class/RT \
252         /usr/lib/class/SDC \
253         /usr/lib/class/TS \
254         /usr/lib/crypto \
```

```
255         /usr/lib/drv \
256         /usr/lib/elfedit \
257         /usr/lib/fm \
258         /usr/lib/font \
259         /usr/lib/fs \
260         /usr/lib/fs/nfs \
261         /usr/lib/fs/proc \
262         /usr/lib/fs/smb \
263         /usr/lib/fs/zfs \
264         /usr/lib/gss \
265         /usr/lib/hal \
266         /usr/lib/inet \
267         /usr/lib/inet/dhcp \
268         /usr/lib/inet/dhcp/nsu \
269         /usr/lib/inet/dhcp/svc \
270         /usr/lib/inet/dhcp/svcadm \
271         /usr/lib/inet/ilb \
272         /usr/lib/inet/$(MACH32) \
273         /usr/lib/inet/wanboot \
278         $(XDIRS) \
274         /usr/lib/krb5 \
275         /usr/lib/link_audit \
276         /usr/lib/libp \
277         /usr/lib/lwp \
278         /usr/lib/mdb \
279         /usr/lib/mdb/kvm \
280         /usr/lib/mdb/proc \
281         /usr/lib/nfs \
282         /usr/net \
283         /usr/net/servers \
284         /usr/lib/pool \
285         /usr/lib/python2.6 \
286         /usr/lib/python2.6/vendor-packages \
287         /usr/lib/python2.6/vendor-packages/64 \
288         /usr/lib/python2.6/vendor-packages/solaris \
289         /usr/lib/python2.6/vendor-packages/zfs \
290         /usr/lib/python2.6/vendor-packages/beadm \
291         /usr/lib/rcap \
292         /usr/lib/rcap/$(MACH32) \
293         /usr/lib/sa  \
294         /usr/lib/saf \
295         /usr/lib/sasl \
296         /usr/lib/scsi \
297         /usr/lib/secure \
298         /usr/lib/security \
299         /usr/lib/smbsrv \
300         /usr/lib/vscan \
301         /usr/lib/zfs \
302         /usr/lib/zones \
303         /usr/old \
304         /usr/platform  \
305         /usr/proc \
306         /usr/proc/bin \
307         /usr/sadm \
308         /usr/sadm/install \
309         /usr/sadm/install/bin \
310         /usr/sadm/install/scripts \
311         /usr/sbin \
312         /usr/sbin/$(MACH32) \
313         /usr/share  \
314         /usr/share/applications \
315         /usr/share/audio \
316         /usr/share/audio/samples \
317         /usr/share/audio/samples/au \
318         /usr/share/gnome \
319         /usr/share/gnome/autostart \
```



```
320         /usr/share/hwdata \
321         /usr/share/lib \
322         /usr/share/lib/ccs \
323         /usr/share/lib/tmac  \
324         /usr/share/lib/ldif  \
325         /usr/share/lib/xml \
326         /usr/share/lib/xml/dtd \
327         /usr/share/man \
328         /usr/share/man/man1 \
329         /usr/share/man/man1b \
330         /usr/share/man/man1c \
331         /usr/share/man/man1m \
332         /usr/share/man/man2 \
333         /usr/share/man/man3 \
334         /usr/share/man/man3bsm \
335         /usr/share/man/man3c \
336         /usr/share/man/man3c_db \
337         /usr/share/man/man3cfgadm \
338         /usr/share/man/man3commputil \
339         /usr/share/man/man3contract \
340         /usr/share/man/man3cpc \
341         /usr/share/man/man3curses \
342         /usr/share/man/man3dat \
343         /usr/share/man/man3devid \
344         /usr/share/man/man3devinfo \
345         /usr/share/man/man3dlpi \
346         /usr/share/man/man3dns_sd \
347         /usr/share/man/man3elf \
348         /usr/share/man/man3exacct \
349         /usr/share/man/man3ext \
350         /usr/share/man/man3fcoe \
351         /usr/share/man/man3fstyp \
352         /usr/share/man/man3gen \
353         /usr/share/man/man3gss \
354         /usr/share/man/man3head \
355         /usr/share/man/man3iscsit \
356         /usr/share/man/man3kstat \
357         /usr/share/man/man3kvm \
358         /usr/share/man/man3ldap \
359         /usr/share/man/man3lgrp \
360         /usr/share/man/man3lib \
361         /usr/share/man/man3libucb \
362         /usr/share/man/man3mail \
363         /usr/share/man/man3malloc \
364         /usr/share/man/man3mp \
365         /usr/share/man/man3mpapi \
366         /usr/share/man/man3nsl \
367         /usr/share/man/man3nvpair \
368         /usr/share/man/man3pam \
369         /usr/share/man/man3papi \
370         /usr/share/man/man3perl \
371         /usr/share/man/man3picl \
372         /usr/share/man/man3picltree \
373         /usr/share/man/man3pool \
374         /usr/share/man/man3proc \
375         /usr/share/man/man3project \
376         /usr/share/man/man3resolv \
377         /usr/share/man/man3rpc \
378         /usr/share/man/man3rsm \
379         /usr/share/man/man3sasl \
380         /usr/share/man/man3scf \
381         /usr/share/man/man3sec \
382         /usr/share/man/man3secdb \
383         /usr/share/man/man3sip \
384         /usr/share/man/man3slp \
385         /usr/share/man/man3socket \
```

```
386         /usr/share/man/man3stmf \
387         /usr/share/man/man3sysevent \
388         /usr/share/man/man3tecla \
389         /usr/share/man/man3tnf \
390         /usr/share/man/man3tsol \
391         /usr/share/man/man3ucb \
392         /usr/share/man/man3uuid \
393         /usr/share/man/man3volmgt \
394         /usr/share/man/man3xcurses \
395         /usr/share/man/man3xnet \
396         /usr/share/man/man4 \
397         /usr/share/man/man5 \
398         /usr/share/man/man7 \
399         /usr/share/man/man7d \
400         /usr/share/man/man7fs \
401         /usr/share/man/man7i \
402         /usr/share/man/man7ipp \
403         /usr/share/man/man7m \
404         /usr/share/man/man7p \
405         /usr/share/man/man9 \
406         /usr/share/man/man9e \
407         /usr/share/man/man9f \
408         /usr/share/man/man9p \
409         /usr/share/man/man9s \
410         /usr/share/src \
411         /usr/snadm \
412         /usr/snadm/lib \
413         /usr/ucb \
414         $(MACH32_DIRS) \
415         /usr/ucblib \
416         /usr/xpg4 \
417         /usr/xpg4/bin \
418         /usr/xpg4/include \
419         /usr/xpg4/lib \
420         /usr/xpg6 \
421         /usr/xpg6/bin \
422         /var  \
423         /var/adm  \
424         /var/adm/exacct \
425         /var/adm/log \
426         /var/adm/pool \
427         /var/adm/sa \
428         /var/adm/sm.bin \
429         /var/adm/streams \
430         /var/cores \
431         /var/cron \
432         /var/db \
433         /var/db/ipf \
434         /var/games \
435         /var/idmap \
436         /var/krb5 \
437         /var/krb5/rcache  \
438         /var/krb5/rcache/root  \
439         /var/ld \
440         /var/log \
441         /var/log/pool \
442         /var/logadm \
443         /var/mail \
444         /var/news \
445         /var/opt \
446         /var/preserve \
447         /var/run \
448         /var/saf \
449         /var/sadm \
450         /var/sadm/install \
451         /var/sadm/install/admin \
```



```
452         /var/sadm/install/logs \
453         /var/sadm/pkg \
454         /var/sadm/security \
455         /var/smb \
456         /var/smb/cvol \
457         /var/smb/cvol/windows \
458         /var/smb/cvol/windows/system32 \
459         /var/smb/cvol/windows/system32/vss \
460         /var/spool \
461         /var/spool/cron \
462         /var/spool/cron/atjobs \
463         /var/spool/cron/crontabs \
464         /var/spool/lp \
465         /var/spool/pkg \
466         /var/spool/uucp \
467         /var/spool/uucppublic \
468         /var/svc \
469         /var/svc/log \
470         /var/svc/manifest \
471         /var/svc/manifest/milestone \
472         /var/svc/manifest/device \
473         /var/svc/manifest/system \
474         /var/svc/manifest/system/device \
475         /var/svc/manifest/system/filesystem \
476         /var/svc/manifest/system/security \
477         /var/svc/manifest/system/svc \
478         /var/svc/manifest/network \
479         /var/svc/manifest/network/dns \
480         /var/svc/manifest/network/ipsec \
481         /var/svc/manifest/network/ldap \
482         /var/svc/manifest/network/nfs \
483         /var/svc/manifest/network/nis \
484         /var/svc/manifest/network/rpc \
485         /var/svc/manifest/network/routing \
486         /var/svc/manifest/network/security \
487         /var/svc/manifest/network/shares \
488         /var/svc/manifest/network/ssl \
489         /var/svc/manifest/application \
490         /var/svc/manifest/application/management \
491         /var/svc/manifest/application/print \
492         /var/svc/manifest/application/security \
493         /var/svc/manifest/platform \
494         /var/svc/manifest/platform/sun4u \
495         /var/svc/manifest/platform/sun4v \
496         /var/svc/manifest/site \
497         /var/svc/profile \
498         /var/uucp \
499         /var/tmp \
500         /var/tsol \
501         /var/tsol/doors

503 sparcv9_DIRS64= \
504         /platform/sun4u \
505         /platform/sun4u/lib \
506         /platform/sun4u/lib/$(MACH64) \
507         /usr/platform/sun4u \
508         /usr/platform/sun4u/sbin \
509         /usr/platform/sun4u/lib \
510         /platform/sun4v/lib \
511         /platform/sun4v/lib/$(MACH64) \
512         /usr/platform/sun4v/sbin \
513         /usr/platform/sun4v/lib \
514         /usr/platform/sun4u-us3/lib \
515         /usr/platform/sun4u-opl/lib

517 amd64_DIRS64= \
```

```
518         /platform/i86pc/amd64

520 DIRS64= \
521         $($(MACH64)_DIRS64) \
522         /lib/$(MACH64) \
523         /lib/crypto/$(MACH64) \
524         /lib/fm/$(MACH64) \
525         /lib/secure/$(MACH64) \
526         /usr/bin/$(MACH64) \
527         /usr/ccs/bin/$(MACH64)  \
528         /usr/ccs/lib/$(MACH64) \
529         /usr/lib/$(MACH64) \
530         /usr/lib/$(MACH64)/gss \
531         /usr/lib/brand/sn1/$(MACH64) \
532         /usr/lib/brand/solaris10/$(MACH64) \
533         /usr/lib/elfedit/$(MACH64) \
534         /usr/lib/fm/$(MACH64) \
535         /usr/lib/fs/nfs/$(MACH64) \
536         /usr/lib/fs/smb/$(MACH64) \
537         /usr/lib/inet/$(MACH64) \
538         /usr/lib/krb5/$(MACH64) \
539         /usr/lib/libp/$(MACH64) \
540         /usr/lib/link_audit/$(MACH64) \
541         /usr/lib/lwp/$(MACH64) \
542         /usr/lib/mdb/kvm/$(MACH64) \
543         /usr/lib/mdb/proc/$(MACH64) \
544         /usr/lib/rcap/$(MACH64) \
545         /usr/lib/sasl/$(MACH64) \
546         /usr/lib/scsi/$(MACH64) \
547         /usr/lib/secure/$(MACH64) \
548         /usr/lib/security/$(MACH64) \
549         /usr/lib/smbsrv/$(MACH64) \
550         /usr/lib/abi/$(MACH64) \
551         /usr/sbin/$(MACH64) \
552         /usr/ucb/$(MACH64) \
553         /usr/ucblib/$(MACH64) \
554         /usr/xpg4/lib/$(MACH64) \
555         /var/ld/$(MACH64)

557 # /var/mail/:saved is built directly by the rootdirs target in
558 # /usr/src/Makefile because of the colon in its name.

560 # macros for symbolic links
561 SYM.DIRS= \
562         /bin \
563         /dev/stdin \
564         /dev/stdout \
565         /dev/stderr \
566         /etc/lib/ld.so.1 \
567         /etc/lib/libdl.so.1 \
568         /etc/lib/nss_files.so.1 \
569         /etc/log \
570         /lib/32 \
571         /lib/crypto/32 \
572         /lib/secure/32 \
573         /usr/adm \
574         /usr/spool \
575         /usr/lib/tmac \
576         /usr/ccs/lib/link_audit \
577         /usr/news \
578         /usr/preserve \
579         /usr/lib/32 \
580         /usr/lib/cron \
581         /usr/lib/elfedit/32 \
582         /usr/lib/libp/32 \
583         /usr/lib/lwp/32 \
```



```
584         /usr/lib/link_audit/32 \
585         /usr/lib/secure/32 \
586         /usr/mail \
587         /usr/man \
588         /usr/pub \
589         /usr/src \
590         /usr/tmp \
591         /usr/ucblib/32 \
592         /var/ld/32

594 sparc_SYM.DIRS64=

596 SYM.DIRS64= \
597         $($(MACH)_SYM.DIRS64) \
598         /lib/64 \
599         /lib/crypto/64 \
600         /lib/secure/64 \
601         /usr/lib/64 \
602         /usr/lib/brand/sn1/64 \
603         /usr/lib/brand/solaris10/64 \
604         /usr/lib/elfedit/64 \
605         /usr/lib/libp/64 \
606         /usr/lib/link_audit/64 \
607         /usr/lib/lwp/64 \
608         /usr/lib/secure/64 \
609         /usr/lib/security/64 \
610         /usr/xpg4/lib/64 \
611         /var/ld/64 \
612         /usr/ucblib/64

614 # prepend the ROOT prefix

616 ROOTDIRS=       $(TARGETDIRS:%=$(ROOT)%)

618 # conditional assignments
619 #
620 # Target directories with non-default values for owner and group must
621 # be referenced here, using their fully-prefixed names, and the non-
622 # default values assigned.  If a directory is mentioned above and not
623 # mentioned below, it has default values for attributes.
624 #
625 # The default value for DIRMODE is specified in usr/src/Makefile.master.
626 #

628 $(ROOT)/var/adm \
629 $(ROOT)/var/adm/sa :=           DIRMODE= 775

631 $(ROOT)/var/spool/lp:=          DIRMODE= 775

633 # file mode
634 #
635 $(ROOT)/tmp \
636 $(ROOT)/var/krb5/rcache \
637 $(ROOT)/var/preserve \
638 $(ROOT)/var/spool/pkg \
639 $(ROOT)/var/spool/uucppublic \
640 $(ROOT)/var/tmp:=       DIRMODE= 1777

642 $(ROOT)/root:=          DIRMODE= 700

644 $(ROOT)/var/krb5/rcache/root:=  DIRMODE= 700


647 #
648 # These permissions must match those set
649 # in the package manifests.
```

```
650 #
651 $(ROOT)/var/sadm/pkg \
652 $(ROOT)/var/sadm/security \
653 $(ROOT)/var/sadm/install/logs :=        DIRMODE= 555


656 #
657 # These permissions must match the ones set
658 # internally by fdfs and autofs.
659 #
660 $(ROOT)/dev/fd \
661 $(ROOT)/home:=          DIRMODE= 555

663 $(ROOT)/var/mail:=      DIRMODE=1777

665 $(ROOT)/proc:=          DIRMODE= 555

667 $(ROOT)/system/contract:=       DIRMODE= 555
668 $(ROOT)/system/object:=         DIRMODE= 555

670 # symlink assignments, LINKDEST is the value of the symlink
671 #
672 $(ROOT)/usr/lib/cron:=                  LINKDEST=../../etc/cron.d
673 $(ROOT)/bin:=                           LINKDEST=usr/bin
674 $(ROOT)/lib/32:=                        LINKDEST=.
675 $(ROOT)/lib/crypto/32:=                 LINKDEST=.
676 $(ROOT)/lib/secure/32:=                 LINKDEST=.
677 $(ROOT)/dev/stdin:=                     LINKDEST=fd/0
678 $(ROOT)/dev/stdout:=                    LINKDEST=fd/1
679 $(ROOT)/dev/stderr:=                    LINKDEST=fd/2
680 $(ROOT)/usr/pub:=                       LINKDEST=share/lib/pub
681 $(ROOT)/usr/man:=                       LINKDEST=share/man
682 $(ROOT)/usr/src:=                       LINKDEST=share/src
683 $(ROOT)/usr/adm:=                       LINKDEST=../var/adm
684 $(ROOT)/etc/lib/ld.so.1:=               LINKDEST=../../lib/ld.so.1
685 $(ROOT)/etc/lib/libdl.so.1:=            LINKDEST=../../lib/libdl.so.1
686 $(ROOT)/etc/lib/nss_files.so.1:=        LINKDEST=../../lib/nss_files.so.1
687 $(ROOT)/etc/log:=                       LINKDEST=../var/adm/log
688 $(ROOT)/usr/mail:=                      LINKDEST=../var/mail
689 $(ROOT)/usr/news:=                      LINKDEST=../var/news
690 $(ROOT)/usr/preserve:=                  LINKDEST=../var/preserve
691 $(ROOT)/usr/spool:=                     LINKDEST=../var/spool
692 $(ROOT)/usr/tmp:=                       LINKDEST=../var/tmp
693 $(ROOT)/usr/lib/tmac:=                  LINKDEST=../share/lib/tmac
694 $(ROOT)/usr/lib/32:=                    LINKDEST=.
695 $(ROOT)/usr/lib/elfedit/32:=            LINKDEST=.
696 $(ROOT)/usr/lib/libp/32:=               LINKDEST=.
697 $(ROOT)/usr/lib/lwp/32:=                LINKDEST=.
698 $(ROOT)/usr/lib/link_audit/32:=         LINKDEST=.
699 $(ROOT)/usr/lib/secure/32:=             LINKDEST=.
700 $(ROOT)/usr/ccs/lib/link_audit:=        LINKDEST=../../lib/link_audit
701 $(ROOT)/var/ld/32:=                     LINKDEST=.
702 $(ROOT)/usr/ucblib/32:=                 LINKDEST=.


705 $(BUILD64) $(ROOT)/lib/64:=            LINKDEST=$(MACH64)
706 $(BUILD64) $(ROOT)/lib/crypto/64:=      LINKDEST=$(MACH64)
707 $(BUILD64) $(ROOT)/lib/secure/64:=      LINKDEST=$(MACH64)
708 $(BUILD64) $(ROOT)/usr/lib/64:=         LINKDEST=$(MACH64)
709 $(BUILD64) $(ROOT)/usr/lib/elfedit/64:= LINKDEST=$(MACH64)
710 $(BUILD64) $(ROOT)/usr/lib/brand/sn1/64:=       LINKDEST=$(MACH64)
711 $(BUILD64) $(ROOT)/usr/lib/brand/solaris10/64:= LINKDEST=$(MACH64)
712 $(BUILD64) $(ROOT)/usr/lib/libp/64:=    LINKDEST=$(MACH64)
713 $(BUILD64) $(ROOT)/usr/lib/lwp/64:=     LINKDEST=$(MACH64)
714 $(BUILD64) $(ROOT)/usr/lib/link_audit/64:=      LINKDEST=$(MACH64)
715 $(BUILD64) $(ROOT)/usr/lib/secure/64:=  LINKDEST=$(MACH64)
```



```
716 $(BUILD64) $(ROOT)/usr/lib/security/64:=        LINKDEST=$(MACH64)
717 $(BUILD64) $(ROOT)/usr/xpg4/lib/64:=    LINKDEST=$(MACH64)
718 $(BUILD64) $(ROOT)/var/ld/64:=          LINKDEST=$(MACH64)
719 $(BUILD64) $(ROOT)/usr/ucblib/64:=      LINKDEST=$(MACH64)

721 #
722 # Installing a directory symlink calls for overriding INS.dir to install
723 # a symlink.
724 #
725 $(DIRLINKS:%=$(ROOT)%):= \
726     INS.dir= -$(RM) -r $@; $(SYMLINK) $(LINKDEST) $@

728 # Special symlinks to populate usr/ccs/lib, whose objects
729 # have actually been moved to usr/lib
730 # Rather than adding another set of rules, we add usr/lib/lwp files here
731 $(ROOT)/usr/ccs/lib/libcurses.so:=      REALPATH=../../../lib/libcurses.so.1
732 $(ROOT)/usr/ccs/lib/llib-lcurses:=      REALPATH=../../../lib/llib-lcurses
733 $(ROOT)/usr/ccs/lib/llib-lcurses.ln:=   REALPATH=../../../lib/llib-lcurses.ln
734 $(ROOT)/usr/ccs/lib/libform.so:=        REALPATH=../../lib/libform.so.1
735 $(ROOT)/usr/ccs/lib/llib-lform:=        REALPATH=../../lib/llib-lform
736 $(ROOT)/usr/ccs/lib/llib-lform.ln:=     REALPATH=../../lib/llib-lform.ln
737 $(ROOT)/usr/ccs/lib/libgen.so:=         REALPATH=../../../lib/libgen.so.1
738 $(ROOT)/usr/ccs/lib/llib-lgen:=         REALPATH=../../../lib/llib-lgen
739 $(ROOT)/usr/ccs/lib/llib-lgen.ln:=      REALPATH=../../../lib/llib-lgen.ln
740 $(ROOT)/usr/ccs/lib/libmalloc.so:=      REALPATH=../../lib/libmalloc.so.1
741 $(ROOT)/usr/ccs/lib/libmenu.so:=        REALPATH=../../lib/libmenu.so.1
742 $(ROOT)/usr/ccs/lib/llib-lmenu:=        REALPATH=../../lib/llib-lmenu
743 $(ROOT)/usr/ccs/lib/llib-lmenu.ln:=     REALPATH=../../lib/llib-lmenu.ln
744 $(ROOT)/usr/ccs/lib/libpanel.so:=       REALPATH=../../lib/libpanel.so.1
745 $(ROOT)/usr/ccs/lib/llib-lpanel:=       REALPATH=../../lib/llib-lpanel
746 $(ROOT)/usr/ccs/lib/llib-lpanel.ln:=    REALPATH=../../lib/llib-lpanel.ln
747 $(ROOT)/usr/ccs/lib/libtermlib.so:=     REALPATH=../../../lib/libcurses.so.1
748 $(ROOT)/usr/ccs/lib/llib-ltermlib:=     REALPATH=../../../lib/llib-lcurses
749 $(ROOT)/usr/ccs/lib/llib-ltermlib.ln:=  REALPATH=../../../lib/llib-lcurses.ln
750 $(ROOT)/usr/ccs/lib/libtermcap.so:=     REALPATH=../../../lib/libtermcap.so.1
751 $(ROOT)/usr/ccs/lib/llib-ltermcap:=     REALPATH=../../../lib/llib-ltermcap
752 $(ROOT)/usr/ccs/lib/llib-ltermcap.ln:=  REALPATH=../../../lib/llib-ltermcap.ln
753 $(ROOT)/usr/ccs/lib/values-Xa.o:=       REALPATH=../../lib/values-Xa.o
754 $(ROOT)/usr/ccs/lib/values-Xc.o:=       REALPATH=../../lib/values-Xc.o
755 $(ROOT)/usr/ccs/lib/values-Xs.o:=       REALPATH=../../lib/values-Xs.o
756 $(ROOT)/usr/ccs/lib/values-Xt.o:=       REALPATH=../../lib/values-Xt.o
757 $(ROOT)/usr/ccs/lib/values-xpg4.o:=     REALPATH=../../lib/values-xpg4.o
758 $(ROOT)/usr/ccs/lib/values-xpg6.o:=     REALPATH=../../lib/values-xpg6.o
759 $(ROOT)/usr/ccs/lib/libl.so:=           REALPATH=../../lib/libl.so.1
760 $(ROOT)/usr/ccs/lib/llib-ll.ln:=        REALPATH=../../lib/llib-ll.ln
761 $(ROOT)/usr/ccs/lib/liby.so:=           REALPATH=../../lib/liby.so.1
762 $(ROOT)/usr/ccs/lib/llib-ly.ln:=        REALPATH=../../lib/llib-ly.ln
763 $(ROOT)/usr/lib/libp/libc.so.1:=        REALPATH=../../../lib/libc.so.1
764 $(ROOT)/usr/lib/lwp/libthread.so.1:=    REALPATH=../libthread.so.1
765 $(ROOT)/usr/lib/lwp/libthread_db.so.1:= REALPATH=../libthread_db.so.1

767 # symlinks to populate usr/ccs/lib/$(MACH64)
768 $(ROOT)/usr/ccs/lib/$(MACH64)/libcurses.so:= \
769         REALPATH=../../../../lib/$(MACH64)/libcurses.so.1
770 $(ROOT)/usr/ccs/lib/$(MACH64)/llib-lcurses.ln:= \
771         REALPATH=../../../../lib/$(MACH64)/llib-lcurses.ln
772 $(ROOT)/usr/ccs/lib/$(MACH64)/libform.so:= \
773         REALPATH=../../../lib/$(MACH64)/libform.so.1
774 $(ROOT)/usr/ccs/lib/$(MACH64)/llib-lform.ln:= \
775         REALPATH=../../../lib/$(MACH64)/llib-lform.ln
776 $(ROOT)/usr/ccs/lib/$(MACH64)/libgen.so:= \
777         REALPATH=../../../../lib/$(MACH64)/libgen.so.1
778 $(ROOT)/usr/ccs/lib/$(MACH64)/llib-lgen.ln:= \
779         REALPATH=../../../../lib/$(MACH64)/llib-lgen.ln
780 $(ROOT)/usr/ccs/lib/$(MACH64)/libmalloc.so:= \
781         REALPATH=../../../lib/$(MACH64)/libmalloc.so.1
```

```
$(ROOT)/usr/ccs/lib/$(MACH64)/libmenu.so:= \
        REALPATH=../../../lib/$(MACH64)/libmenu.so.1
$(ROOT)/usr/ccs/lib/$(MACH64)/llib-lmenu.ln:= \
        REALPATH=../../../lib/$(MACH64)/llib-lmenu.ln
$(ROOT)/usr/ccs/lib/$(MACH64)/libpanel.so:= \
        REALPATH=../../../lib/$(MACH64)/libpanel.so.1
$(ROOT)/usr/ccs/lib/$(MACH64)/llib-lpanel.ln:= \
        REALPATH=../../../lib/$(MACH64)/llib-lpanel.ln
$(ROOT)/usr/ccs/lib/$(MACH64)/libtermlib.so:= \
        REALPATH=../../../../lib/$(MACH64)/libcurses.so.1
$(ROOT)/usr/ccs/lib/$(MACH64)/llib-ltermlib.ln:= \
        REALPATH=../../../../lib/$(MACH64)/llib-lcurses.ln
$(ROOT)/usr/ccs/lib/$(MACH64)/libtermcap.so:= \
        REALPATH=../../../../lib/$(MACH64)/libtermcap.so.1
$(ROOT)/usr/ccs/lib/$(MACH64)/llib-ltermcap.ln:= \
        REALPATH=../../../../lib/$(MACH64)/llib-ltermcap.ln
$(ROOT)/usr/ccs/lib/$(MACH64)/values-Xa.o:= \
        REALPATH=../../../lib/$(MACH64)/values-Xa.o
$(ROOT)/usr/ccs/lib/$(MACH64)/values-Xc.o:= \
        REALPATH=../../../lib/$(MACH64)/values-Xc.o
$(ROOT)/usr/ccs/lib/$(MACH64)/values-Xs.o:= \
        REALPATH=../../../lib/$(MACH64)/values-Xs.o
$(ROOT)/usr/ccs/lib/$(MACH64)/values-Xt.o:= \
        REALPATH=../../../lib/$(MACH64)/values-Xt.o
$(ROOT)/usr/ccs/lib/$(MACH64)/values-xpg4.o:= \
        REALPATH=../../../lib/$(MACH64)/values-xpg4.o
$(ROOT)/usr/ccs/lib/$(MACH64)/values-xpg6.o:= \
        REALPATH=../../../lib/$(MACH64)/values-xpg6.o
$(ROOT)/usr/ccs/lib/$(MACH64)/libl.so:= \
        REALPATH=../../../lib/$(MACH64)/libl.so.1
$(ROOT)/usr/ccs/lib/$(MACH64)/llib-ll.ln:= \
        REALPATH=../../../lib/$(MACH64)/llib-ll.ln
$(ROOT)/usr/ccs/lib/$(MACH64)/liby.so:= \
        REALPATH=../../../lib/$(MACH64)/liby.so.1
$(ROOT)/usr/ccs/lib/$(MACH64)/llib-ly.ln:= \
        REALPATH=../../../lib/$(MACH64)/llib-ly.ln
$(ROOT)/usr/lib/libp/$(MACH64)/libc.so.1:= \
        REALPATH=../../../../lib/$(MACH64)/libc.so.1
$(ROOT)/usr/lib/lwp/$(MACH64)/libthread.so.1:= \
        REALPATH=../../$(MACH64)/libthread.so.1
$(ROOT)/usr/lib/lwp/$(MACH64)/libthread_db.so.1:= \
        REALPATH=../../$(MACH64)/libthread_db.so.1

SYM.USRCCSLIB= \
        /usr/ccs/lib/libcurses.so \
        /usr/ccs/lib/llib-lcurses \
        /usr/ccs/lib/llib-lcurses.ln \
        /usr/ccs/lib/libform.so \
        /usr/ccs/lib/llib-lform \
        /usr/ccs/lib/llib-lform.ln \
        /usr/ccs/lib/libgen.so \
        /usr/ccs/lib/llib-lgen \
        /usr/ccs/lib/llib-lgen.ln \
        /usr/ccs/lib/libmalloc.so \
        /usr/ccs/lib/libmenu.so \
        /usr/ccs/lib/llib-lmenu \
        /usr/ccs/lib/llib-lmenu.ln \
        /usr/ccs/lib/libpanel.so \
        /usr/ccs/lib/llib-lpanel \
        /usr/ccs/lib/llib-lpanel.ln \
        /usr/ccs/lib/libtermlib.so \
        /usr/ccs/lib/llib-ltermlib \
        /usr/ccs/lib/llib-ltermlib.ln \
        /usr/ccs/lib/libtermcap.so \
        /usr/ccs/lib/llib-ltermcap \
        /usr/ccs/lib/llib-ltermcap.ln \
```



```
        /usr/ccs/lib/values-Xa.o \
        /usr/ccs/lib/values-Xc.o \
        /usr/ccs/lib/values-Xs.o \
        /usr/ccs/lib/values-Xt.o \
        /usr/ccs/lib/values-xpg4.o \
        /usr/ccs/lib/values-xpg6.o \
        /usr/ccs/lib/libl.so \
        /usr/ccs/lib/llib-ll.ln \
        /usr/ccs/lib/liby.so \
        /usr/ccs/lib/llib-ly.ln \
        /usr/lib/libp/libc.so.1 \
        /usr/lib/lwp/libthread.so.1 \
        /usr/lib/lwp/libthread_db.so.1

SYM.USRCCSLIB64= \
        /usr/ccs/lib/$(MACH64)/libcurses.so \
        /usr/ccs/lib/$(MACH64)/llib-lcurses.ln \
        /usr/ccs/lib/$(MACH64)/libform.so \
        /usr/ccs/lib/$(MACH64)/llib-lform.ln \
        /usr/ccs/lib/$(MACH64)/libgen.so \
        /usr/ccs/lib/$(MACH64)/llib-lgen.ln \
        /usr/ccs/lib/$(MACH64)/libmalloc.so \
        /usr/ccs/lib/$(MACH64)/libmenu.so \
        /usr/ccs/lib/$(MACH64)/llib-lmenu.ln \
        /usr/ccs/lib/$(MACH64)/libpanel.so \
        /usr/ccs/lib/$(MACH64)/llib-lpanel.ln \
        /usr/ccs/lib/$(MACH64)/libtermlib.so \
        /usr/ccs/lib/$(MACH64)/llib-ltermlib.ln \
        /usr/ccs/lib/$(MACH64)/libtermcap.so \
        /usr/ccs/lib/$(MACH64)/llib-ltermcap.ln \
        /usr/ccs/lib/$(MACH64)/values-Xa.o \
        /usr/ccs/lib/$(MACH64)/values-Xc.o \
        /usr/ccs/lib/$(MACH64)/values-Xs.o \
        /usr/ccs/lib/$(MACH64)/values-Xt.o \
        /usr/ccs/lib/$(MACH64)/values-xpg4.o \
        /usr/ccs/lib/$(MACH64)/values-xpg6.o \
        /usr/ccs/lib/$(MACH64)/libl.so \
        /usr/ccs/lib/$(MACH64)/llib-ll.ln \
        /usr/ccs/lib/$(MACH64)/liby.so \
        /usr/ccs/lib/$(MACH64)/llib-ly.ln \
        /usr/lib/libp/$(MACH64)/libc.so.1 \
        /usr/lib/lwp/$(MACH64)/libthread.so.1 \
        /usr/lib/lwp/$(MACH64)/libthread_db.so.1

# Special symlinks to direct libraries that have been moved
# from /usr/lib to /lib in order to live in the root filesystem.
$(ROOT)/lib/libposix4.so.1:=            REALPATH=librt.so.1
$(ROOT)/lib/libposix4.so:=              REALPATH=libposix4.so.1
$(ROOT)/lib/llib-lposix4:=              REALPATH=llib-lrt
$(ROOT)/lib/llib-lposix4.ln:=           REALPATH=llib-lrt.ln
$(ROOT)/lib/libthread_db.so.1:=         REALPATH=libc_db.so.1
$(ROOT)/lib/libthread_db.so:=           REALPATH=libc_db.so.1
$(ROOT)/usr/lib/ld.so.1:=               REALPATH=../../lib/ld.so.1
$(ROOT)/usr/lib/libadm.so.1:=           REALPATH=../../lib/libadm.so.1
$(ROOT)/usr/lib/libadm.so:=             REALPATH=../../lib/libadm.so.1
$(ROOT)/usr/lib/libaio.so.1:=           REALPATH=../../lib/libaio.so.1
$(ROOT)/usr/lib/libaio.so:=             REALPATH=../../lib/libaio.so.1
$(ROOT)/usr/lib/libavl.so.1:=           REALPATH=../../lib/libavl.so.1
$(ROOT)/usr/lib/libavl.so:=             REALPATH=../../lib/libavl.so.1
$(ROOT)/usr/lib/libbsm.so.1:=           REALPATH=../../lib/libbsm.so.1
$(ROOT)/usr/lib/libbsm.so:=             REALPATH=../../lib/libbsm.so.1
$(ROOT)/usr/lib/libc.so.1:=             REALPATH=../../lib/libc.so.1
$(ROOT)/usr/lib/libc.so:=               REALPATH=../../lib/libc.so.1
$(ROOT)/usr/lib/libc_db.so.1:=          REALPATH=../../lib/libc_db.so.1
$(ROOT)/usr/lib/libc_db.so:=            REALPATH=../../lib/libc_db.so.1
$(ROOT)/usr/lib/libcmdutils.so.1:=      REALPATH=../../lib/libcmdutils.so.1
```

```
914 $(ROOT)/usr/lib/libcmdutils.so:=        REALPATH=../../lib/libcmdutils.so.1
915 $(ROOT)/usr/lib/libcontract.so.1:=      REALPATH=../../lib/libcontract.so.1
916 $(ROOT)/usr/lib/libcontract.so:=        REALPATH=../../lib/libcontract.so.1
917 $(ROOT)/usr/lib/libcryptoutil.so.1:=    REALPATH=../../lib/libcryptoutil.so.1
918 $(ROOT)/usr/lib/libcryptoutil.so:=      REALPATH=../../lib/libcryptoutil.so.1
919 $(ROOT)/usr/lib/libctf.so.1:=           REALPATH=../../lib/libctf.so.1
920 $(ROOT)/usr/lib/libctf.so:=             REALPATH=../../lib/libctf.so.1
921 $(ROOT)/usr/lib/libcurses.so.1:=        REALPATH=../../lib/libcurses.so.1
922 $(ROOT)/usr/lib/libcurses.so:=          REALPATH=../../lib/libcurses.so.1
923 $(ROOT)/usr/lib/libdevice.so.1:=        REALPATH=../../lib/libdevice.so.1
924 $(ROOT)/usr/lib/libdevice.so:=          REALPATH=../../lib/libdevice.so.1
925 $(ROOT)/usr/lib/libdevid.so.1:=         REALPATH=../../lib/libdevid.so.1
926 $(ROOT)/usr/lib/libdevid.so:=           REALPATH=../../lib/libdevid.so.1
927 $(ROOT)/usr/lib/libdevinfo.so.1:=       REALPATH=../../lib/libdevinfo.so.1
928 $(ROOT)/usr/lib/libdevinfo.so:=         REALPATH=../../lib/libdevinfo.so.1
929 $(ROOT)/usr/lib/libdhcpagent.so.1:=     REALPATH=../../lib/libdhcpagent.so.1
930 $(ROOT)/usr/lib/libdhcpagent.so:=       REALPATH=../../lib/libdhcpagent.so.1
931 $(ROOT)/usr/lib/libdhcputil.so.1:=      REALPATH=../../lib/libdhcputil.so.1
932 $(ROOT)/usr/lib/libdhcputil.so:=        REALPATH=../../lib/libdhcputil.so.1
933 $(ROOT)/usr/lib/libdl.so.1:=            REALPATH=../../lib/libdl.so.1
934 $(ROOT)/usr/lib/libdl.so:=              REALPATH=../../lib/libdl.so.1
935 $(ROOT)/usr/lib/libdlpi.so.1:=          REALPATH=../../lib/libdlpi.so.1
936 $(ROOT)/usr/lib/libdlpi.so:=            REALPATH=../../lib/libdlpi.so.1
937 $(ROOT)/usr/lib/libdoor.so.1:=          REALPATH=../../lib/libdoor.so.1
938 $(ROOT)/usr/lib/libdoor.so:=            REALPATH=../../lib/libdoor.so.1
939 $(ROOT)/usr/lib/libefi.so.1:=           REALPATH=../../lib/libefi.so.1
940 $(ROOT)/usr/lib/libefi.so:=             REALPATH=../../lib/libefi.so.1
941 $(ROOT)/usr/lib/libelf.so.1:=           REALPATH=../../lib/libelf.so.1
942 $(ROOT)/usr/lib/libelf.so:=             REALPATH=../../lib/libelf.so.1
943 $(ROOT)/usr/lib/libfdisk.so.1:=         REALPATH=../../lib/libfdisk.so.1
944 $(ROOT)/usr/lib/libfdisk.so:=           REALPATH=../../lib/libfdisk.so.1
945 $(ROOT)/usr/lib/libgen.so.1:=           REALPATH=../../lib/libgen.so.1
946 $(ROOT)/usr/lib/libgen.so:=             REALPATH=../../lib/libgen.so.1
947 $(ROOT)/usr/lib/libinetutil.so.1:=      REALPATH=../../lib/libinetutil.so.1
948 $(ROOT)/usr/lib/libinetutil.so:=        REALPATH=../../lib/libinetutil.so.1
949 $(ROOT)/usr/lib/libintl.so.1:=          REALPATH=../../lib/libintl.so.1
950 $(ROOT)/usr/lib/libintl.so:=            REALPATH=../../lib/libintl.so.1
951 $(ROOT)/usr/lib/libkmf.so.1:=           REALPATH=../../lib/libkmf.so.1
952 $(ROOT)/usr/lib/libkmf.so:=             REALPATH=../../lib/libkmf.so.1
953 $(ROOT)/usr/lib/libkmfberder.so.1:=     REALPATH=../../lib/libkmfberder.so.1
954 $(ROOT)/usr/lib/libkmfberder.so:=       REALPATH=../../lib/libkmfberder.so.1
955 $(ROOT)/usr/lib/libkstat.so.1:=         REALPATH=../../lib/libkstat.so.1
956 $(ROOT)/usr/lib/libkstat.so:=           REALPATH=../../lib/libkstat.so.1
957 $(ROOT)/usr/lib/liblddbg.so.4:=         REALPATH=../../lib/liblddbg.so.4
958 $(ROOT)/usr/lib/libmd.so.1:=            REALPATH=../../lib/libmd.so.1
959 $(ROOT)/usr/lib/libmd.so:=              REALPATH=../../lib/libmd.so.1
960 $(ROOT)/usr/lib/libmd5.so.1:=           REALPATH=../../lib/libmd5.so.1
961 $(ROOT)/usr/lib/libmd5.so:=             REALPATH=../../lib/libmd5.so.1
962 $(ROOT)/usr/lib/libmeta.so.1:=          REALPATH=../../lib/libmeta.so.1
963 $(ROOT)/usr/lib/libmeta.so:=            REALPATH=../../lib/libmeta.so.1
964 $(ROOT)/usr/lib/libmp.so.1:=            REALPATH=../../lib/libmp.so.1
965 $(ROOT)/usr/lib/libmp.so.2:=            REALPATH=../../lib/libmp.so.2
966 $(ROOT)/usr/lib/libmp.so:=              REALPATH=../../lib/libmp.so.2
967 $(ROOT)/usr/lib/libnsl.so.1:=           REALPATH=../../lib/libnsl.so.1
968 $(ROOT)/usr/lib/libnsl.so:=             REALPATH=../../lib/libnsl.so.1
969 $(ROOT)/usr/lib/libnvpair.so.1:=        REALPATH=../../lib/libnvpair.so.1
970 $(ROOT)/usr/lib/libnvpair.so:=          REALPATH=../../lib/libnvpair.so.1
971 $(ROOT)/usr/lib/libpam.so.1:=           REALPATH=../../lib/libpam.so.1
972 $(ROOT)/usr/lib/libpam.so:=             REALPATH=../../lib/libpam.so.1
973 $(ROOT)/usr/lib/libposix4.so.1:=        REALPATH=../../lib/librt.so.1
974 $(ROOT)/usr/lib/libposix4.so:=          REALPATH=../../lib/librt.so.1
975 $(ROOT)/usr/lib/libproc.so.1:=          REALPATH=../../lib/libproc.so.1
976 $(ROOT)/usr/lib/libproc.so:=            REALPATH=../../lib/libproc.so.1
977 $(ROOT)/usr/lib/libpthread.so.1:=       REALPATH=../../lib/libpthread.so.1
978 $(ROOT)/usr/lib/libpthread.so:=         REALPATH=../../lib/libpthread.so.1
979 $(ROOT)/usr/lib/librcm.so.1:=           REALPATH=../../lib/librcm.so.1
```



```
980 $(ROOT)/usr/lib/librcm.so:=            REALPATH=../../lib/librcm.so.1
981 $(ROOT)/usr/lib/libresolv.so.1:=       REALPATH=../../lib/libresolv.so.1
982 $(ROOT)/usr/lib/libresolv.so.2:=       REALPATH=../../lib/libresolv.so.2
983 $(ROOT)/usr/lib/libresolv.so:=         REALPATH=../../lib/libresolv.so.2
984 $(ROOT)/usr/lib/librestart.so.1:=      REALPATH=../../lib/librestart.so.1
985 $(ROOT)/usr/lib/librestart.so:=        REALPATH=../../lib/librestart.so.1
986 $(ROOT)/usr/lib/librpcsvc.so.1:=       REALPATH=../../lib/librpcsvc.so.1
987 $(ROOT)/usr/lib/librpcsvc.so:=         REALPATH=../../lib/librpcsvc.so.1
988 $(ROOT)/usr/lib/librt.so.1:=           REALPATH=../../lib/librt.so.1
989 $(ROOT)/usr/lib/librt.so:=             REALPATH=../../lib/librt.so.1
990 $(ROOT)/usr/lib/librtld.so.1:=         REALPATH=../../lib/librtld.so.1
991 $(ROOT)/usr/lib/librtld_db.so.1:=      REALPATH=../../lib/librtld_db.so.1
992 $(ROOT)/usr/lib/librtld_db.so:=        REALPATH=../../lib/librtld_db.so.1
993 $(ROOT)/usr/lib/libscf.so.1:=          REALPATH=../../lib/libscf.so.1
994 $(ROOT)/usr/lib/libscf.so:=            REALPATH=../../lib/libscf.so.1
995 $(ROOT)/usr/lib/libsec.so.1:=          REALPATH=../../lib/libsec.so.1
996 $(ROOT)/usr/lib/libsec.so:=            REALPATH=../../lib/libsec.so.1
997 $(ROOT)/usr/lib/libsecdb.so.1:=        REALPATH=../../lib/libsecdb.so.1
998 $(ROOT)/usr/lib/libsecdb.so:=          REALPATH=../../lib/libsecdb.so.1
999 $(ROOT)/usr/lib/libsendfile.so.1:=     REALPATH=../../lib/libsendfile.so.1
1000 $(ROOT)/usr/lib/libsendfile.so:=      REALPATH=../../lib/libsendfile.so.1
1001 $(ROOT)/usr/lib/libsocket.so.1:=      REALPATH=../../lib/libsocket.so.1
1002 $(ROOT)/usr/lib/libsocket.so:=        REALPATH=../../lib/libsocket.so.1
1003 $(ROOT)/usr/lib/libsysevent.so.1:=    REALPATH=../../lib/libsysevent.so.1
1004 $(ROOT)/usr/lib/libsysevent.so:=      REALPATH=../../lib/libsysevent.so.1
1005 $(ROOT)/usr/lib/libtermcap.so.1:=     REALPATH=../../lib/libtermcap.so.1
1006 $(ROOT)/usr/lib/libtermcap.so:=       REALPATH=../../lib/libtermcap.so.1
1007 $(ROOT)/usr/lib/libtermlib.so.1:=     REALPATH=../../lib/libcurses.so.1
1008 $(ROOT)/usr/lib/libtermlib.so:=       REALPATH=../../lib/libcurses.so.1
1009 $(ROOT)/usr/lib/libthread.so.1:=      REALPATH=../../lib/libthread.so.1
1010 $(ROOT)/usr/lib/libthread.so:=        REALPATH=../../lib/libthread.so.1
1011 $(ROOT)/usr/lib/libthread_db.so.1:=   REALPATH=../../lib/libc_db.so.1
1012 $(ROOT)/usr/lib/libthread_db.so:=     REALPATH=../../lib/libc_db.so.1
1013 $(ROOT)/usr/lib/libtsnet.so.1:=       REALPATH=../../lib/libtsnet.so.1
1014 $(ROOT)/usr/lib/libtsnet.so:=         REALPATH=../../lib/libtsnet.so.1
1015 $(ROOT)/usr/lib/libtsol.so.2:=        REALPATH=../../lib/libtsol.so.2
1016 $(ROOT)/usr/lib/libtsol.so:=          REALPATH=../../lib/libtsol.so.2
1017 $(ROOT)/usr/lib/libumem.so.1:=        REALPATH=../../lib/libumem.so.1
1018 $(ROOT)/usr/lib/libumem.so:=          REALPATH=../../lib/libumem.so.1
1019 $(ROOT)/usr/lib/libuuid.so.1:=        REALPATH=../../lib/libuuid.so.1
1020 $(ROOT)/usr/lib/libuuid.so:=          REALPATH=../../lib/libuuid.so.1
1021 $(ROOT)/usr/lib/libuutil.so.1:=       REALPATH=../../lib/libuutil.so.1
1022 $(ROOT)/usr/lib/libuutil.so:=         REALPATH=../../lib/libuutil.so.1
1023 $(ROOT)/usr/lib/libw.so.1:=           REALPATH=../../lib/libw.so.1
1024 $(ROOT)/usr/lib/libw.so:=             REALPATH=../../lib/libw.so.1
1025 $(ROOT)/usr/lib/libxnet.so.1:=        REALPATH=../../lib/libxnet.so.1
1026 $(ROOT)/usr/lib/libxnet.so:=          REALPATH=../../lib/libxnet.so.1
1027 $(ROOT)/usr/lib/libzfs.so.1:=         REALPATH=../../lib/libzfs.so.1
1028 $(ROOT)/usr/lib/libzfs.so:=           REALPATH=../../lib/libzfs.so.1
1029 $(ROOT)/usr/lib/libzfs_core.so.1:=    REALPATH=../../lib/libzfs_core.so.1
1030 $(ROOT)/usr/lib/libzfs_core.so:=      REALPATH=../../lib/libzfs_core.so.1
1031 $(ROOT)/usr/lib/llib-ladm.ln:=        REALPATH=../../lib/llib-ladm.ln
1032 $(ROOT)/usr/lib/llib-ladm:=           REALPATH=../../lib/llib-ladm
1033 $(ROOT)/usr/lib/llib-laio.ln:=        REALPATH=../../lib/llib-laio.ln
1034 $(ROOT)/usr/lib/llib-laio:=           REALPATH=../../lib/llib-laio
1035 $(ROOT)/usr/lib/llib-lavl.ln:=        REALPATH=../../lib/llib-lavl.ln
1036 $(ROOT)/usr/lib/llib-lavl:=           REALPATH=../../lib/llib-lavl
1037 $(ROOT)/usr/lib/llib-lbsm.ln:=        REALPATH=../../lib/llib-lbsm.ln
1038 $(ROOT)/usr/lib/llib-lbsm:=           REALPATH=../../lib/llib-lbsm
1039 $(ROOT)/usr/lib/llib-lc.ln:=          REALPATH=../../lib/llib-lc.ln
1040 $(ROOT)/usr/lib/llib-lc:=             REALPATH=../../lib/llib-lc
1041 $(ROOT)/usr/lib/llib-lcmdutils.ln:=   REALPATH=../../lib/llib-lcmdutils.ln
1042 $(ROOT)/usr/lib/llib-lcmdutils:=      REALPATH=../../lib/llib-lcmdutils
1043 $(ROOT)/usr/lib/llib-lcontract.ln:=   REALPATH=../../lib/llib-lcontract.ln
1044 $(ROOT)/usr/lib/llib-lcontract:=      REALPATH=../../lib/llib-lcontract
1045 $(ROOT)/usr/lib/llib-lctf.ln:=        REALPATH=../../lib/llib-lctf.ln
```

```
1046 $(ROOT)/usr/lib/llib-lctf:=             REALPATH=../../lib/llib-lctf
1047 $(ROOT)/usr/lib/llib-lcurses.ln:=       REALPATH=../../lib/llib-lcurses.ln
1048 $(ROOT)/usr/lib/llib-lcurses:=          REALPATH=../../lib/llib-lcurses
1049 $(ROOT)/usr/lib/llib-ldevice.ln:=       REALPATH=../../lib/llib-ldevice.ln
1050 $(ROOT)/usr/lib/llib-ldevice:=          REALPATH=../../lib/llib-ldevice
1051 $(ROOT)/usr/lib/llib-ldevid.ln:=        REALPATH=../../lib/llib-ldevid.ln
1052 $(ROOT)/usr/lib/llib-ldevid:=           REALPATH=../../lib/llib-ldevid
1053 $(ROOT)/usr/lib/llib-ldevinfo.ln:=      REALPATH=../../lib/llib-ldevinfo.ln
1054 $(ROOT)/usr/lib/llib-ldevinfo:=         REALPATH=../../lib/llib-ldevinfo
1055 $(ROOT)/usr/lib/llib-ldhcpagent.ln:=    REALPATH=../../lib/llib-ldhcpagent.ln
1056 $(ROOT)/usr/lib/llib-ldhcpagent:=       REALPATH=../../lib/llib-ldhcpagent
1057 $(ROOT)/usr/lib/llib-ldhcputil.ln:=     REALPATH=../../lib/llib-ldhcputil.ln
1058 $(ROOT)/usr/lib/llib-ldhcputil:=        REALPATH=../../lib/llib-ldhcputil
1059 $(ROOT)/usr/lib/llib-ldl.ln:=           REALPATH=../../lib/llib-ldl.ln
1060 $(ROOT)/usr/lib/llib-ldl:=              REALPATH=../../lib/llib-ldl
1061 $(ROOT)/usr/lib/llib-ldoor.ln:=         REALPATH=../../lib/llib-ldoor.ln
1062 $(ROOT)/usr/lib/llib-ldoor:=            REALPATH=../../lib/llib-ldoor
1063 $(ROOT)/usr/lib/llib-lefi.ln:=          REALPATH=../../lib/llib-lefi.ln
1064 $(ROOT)/usr/lib/llib-lefi:=             REALPATH=../../lib/llib-lefi
1065 $(ROOT)/usr/lib/llib-lelf.ln:=          REALPATH=../../lib/llib-lelf.ln
1066 $(ROOT)/usr/lib/llib-lelf:=             REALPATH=../../lib/llib-lelf
1067 $(ROOT)/usr/lib/llib-lfdisk.ln:=        REALPATH=../../lib/llib-lfdisk.ln
1068 $(ROOT)/usr/lib/llib-lfdisk:=           REALPATH=../../lib/llib-lfdisk
1069 $(ROOT)/usr/lib/llib-lgen.ln:=          REALPATH=../../lib/llib-lgen.ln
1070 $(ROOT)/usr/lib/llib-lgen:=             REALPATH=../../lib/llib-lgen
1071 $(ROOT)/usr/lib/llib-linetutil.ln:=     REALPATH=../../lib/llib-linetutil.ln
1072 $(ROOT)/usr/lib/llib-linetutil:=        REALPATH=../../lib/llib-linetutil
1073 $(ROOT)/usr/lib/llib-lintl.ln:=         REALPATH=../../lib/llib-lintl.ln
1074 $(ROOT)/usr/lib/llib-lintl:=            REALPATH=../../lib/llib-lintl
1075 $(ROOT)/usr/lib/llib-lkstat.ln:=        REALPATH=../../lib/llib-lkstat.ln
1076 $(ROOT)/usr/lib/llib-lkstat:=           REALPATH=../../lib/llib-lkstat
1077 $(ROOT)/usr/lib/llib-lmd5.ln:=          REALPATH=../../lib/llib-lmd5.ln
1078 $(ROOT)/usr/lib/llib-lmd5:=             REALPATH=../../lib/llib-lmd5
1079 $(ROOT)/usr/lib/llib-lmeta.ln:=         REALPATH=../../lib/llib-lmeta.ln
1080 $(ROOT)/usr/lib/llib-lmeta:=            REALPATH=../../lib/llib-lmeta
1081 $(ROOT)/usr/lib/llib-lnsl.ln:=          REALPATH=../../lib/llib-lnsl.ln
1082 $(ROOT)/usr/lib/llib-lnsl:=             REALPATH=../../lib/llib-lnsl
1083 $(ROOT)/usr/lib/llib-lnvpair.ln:=       REALPATH=../../lib/llib-lnvpair.ln
1084 $(ROOT)/usr/lib/llib-lnvpair:=          REALPATH=../../lib/llib-lnvpair
1085 $(ROOT)/usr/lib/llib-lpam.ln:=          REALPATH=../../lib/llib-lpam.ln
1086 $(ROOT)/usr/lib/llib-lpam:=             REALPATH=../../lib/llib-lpam
1087 $(ROOT)/usr/lib/llib-lposix4.ln:=       REALPATH=../../lib/llib-lrt.ln
1088 $(ROOT)/usr/lib/llib-lposix4:=          REALPATH=../../lib/llib-lrt
1089 $(ROOT)/usr/lib/llib-lpthread.ln:=      REALPATH=../../lib/llib-lpthread.ln
1090 $(ROOT)/usr/lib/llib-lpthread:=         REALPATH=../../lib/llib-lpthread
1091 $(ROOT)/usr/lib/llib-lresolv.ln:=       REALPATH=../../lib/llib-lresolv.ln
1092 $(ROOT)/usr/lib/llib-lresolv:=          REALPATH=../../lib/llib-lresolv
1093 $(ROOT)/usr/lib/llib-lrpcsvc.ln:=       REALPATH=../../lib/llib-lrpcsvc.ln
1094 $(ROOT)/usr/lib/llib-lrpcsvc:=          REALPATH=../../lib/llib-lrpcsvc
1095 $(ROOT)/usr/lib/llib-lrt.ln:=           REALPATH=../../lib/llib-lrt.ln
1096 $(ROOT)/usr/lib/llib-lrt:=              REALPATH=../../lib/llib-lrt
1097 $(ROOT)/usr/lib/llib-lrtld_db.ln:=      REALPATH=../../lib/llib-lrtld_db.ln
1098 $(ROOT)/usr/lib/llib-lrtld_db:=         REALPATH=../../lib/llib-lrtld_db
1099 $(ROOT)/usr/lib/llib-lscf.ln:=          REALPATH=../../lib/llib-lscf.ln
1100 $(ROOT)/usr/lib/llib-lscf:=             REALPATH=../../lib/llib-lscf
1101 $(ROOT)/usr/lib/llib-lsec.ln:=          REALPATH=../../lib/llib-lsec.ln
1102 $(ROOT)/usr/lib/llib-lsec:=             REALPATH=../../lib/llib-lsec
1103 $(ROOT)/usr/lib/llib-lsecdb.ln:=        REALPATH=../../lib/llib-lsecdb.ln
1104 $(ROOT)/usr/lib/llib-lsecdb:=           REALPATH=../../lib/llib-lsecdb
1105 $(ROOT)/usr/lib/llib-lsendfile.ln:=     REALPATH=../../lib/llib-lsendfile.ln
1106 $(ROOT)/usr/lib/llib-lsendfile:=        REALPATH=../../lib/llib-lsendfile
1107 $(ROOT)/usr/lib/llib-lsocket.ln:=       REALPATH=../../lib/llib-lsocket.ln
1108 $(ROOT)/usr/lib/llib-lsocket:=          REALPATH=../../lib/llib-lsocket
1109 $(ROOT)/usr/lib/llib-lsysevent.ln:=     REALPATH=../../lib/llib-lsysevent.ln
1110 $(ROOT)/usr/lib/llib-lsysevent:=        REALPATH=../../lib/llib-lsysevent
1111 $(ROOT)/usr/lib/llib-ltermcap.ln:=      REALPATH=../../lib/llib-ltermcap.ln
```



```
1112 $(ROOT)/usr/lib/llib-ltermcap:=         REALPATH=../../lib/llib-ltermcap
1113 $(ROOT)/usr/lib/llib-ltermlib.ln:=      REALPATH=../../lib/llib-lcurses.ln
1114 $(ROOT)/usr/lib/llib-ltermlib:=         REALPATH=../../lib/llib-lcurses
1115 $(ROOT)/usr/lib/llib-lthread.ln:=       REALPATH=../../lib/llib-lthread.ln
1116 $(ROOT)/usr/lib/llib-lthread:=          REALPATH=../../lib/llib-lthread
1117 $(ROOT)/usr/lib/llib-lthread_db.ln:=    REALPATH=../../lib/llib-lc_db.ln
1118 $(ROOT)/usr/lib/llib-lthread_db:=       REALPATH=../../lib/llib-lc_db
1119 $(ROOT)/usr/lib/llib-ltsnet.ln:=        REALPATH=../../lib/llib-ltsnet.ln
1120 $(ROOT)/usr/lib/llib-ltsnet:=           REALPATH=../../lib/llib-ltsnet
1121 $(ROOT)/usr/lib/llib-ltsol.ln:=         REALPATH=../../lib/llib-ltsol.ln
1122 $(ROOT)/usr/lib/llib-ltsol:=            REALPATH=../../lib/llib-ltsol
1123 $(ROOT)/usr/lib/llib-lumem.ln:=         REALPATH=../../lib/llib-lumem.ln
1124 $(ROOT)/usr/lib/llib-lumem:=            REALPATH=../../lib/llib-lumem
1125 $(ROOT)/usr/lib/llib-luuid.ln:=         REALPATH=../../lib/llib-luuid.ln
1126 $(ROOT)/usr/lib/llib-luuid:=            REALPATH=../../lib/llib-luuid
1127 $(ROOT)/usr/lib/llib-lxnet.ln:=         REALPATH=../../lib/llib-lxnet.ln
1128 $(ROOT)/usr/lib/llib-lxnet:=            REALPATH=../../lib/llib-lxnet
1129 $(ROOT)/usr/lib/llib-lzfs.ln:=          REALPATH=../../lib/llib-lzfs.ln
1130 $(ROOT)/usr/lib/llib-lzfs:=             REALPATH=../../lib/llib-lzfs
1131 $(ROOT)/usr/lib/llib-lzfs_core.ln:=     REALPATH=../../lib/llib-lzfs_core.ln
1132 $(ROOT)/usr/lib/llib-lzfs_core:=        REALPATH=../../lib/llib-lzfs_core
1133 $(ROOT)/usr/lib/nss_compat.so.1:=       REALPATH=../../lib/nss_compat.so.1
1134 $(ROOT)/usr/lib/nss_dns.so.1:=          REALPATH=../../lib/nss_dns.so.1
1135 $(ROOT)/usr/lib/nss_files.so.1:=        REALPATH=../../lib/nss_files.so.1
1136 $(ROOT)/usr/lib/nss_nis.so.1:=          REALPATH=../../lib/nss_nis.so.1
1137 $(ROOT)/usr/lib/nss_user.so.1:=         REALPATH=../../lib/nss_user.so.1
1138 $(ROOT)/usr/lib/fm/libfmevent.so.1:=    REALPATH=../../../lib/fm/libfmevent.so.1
1139 $(ROOT)/usr/lib/fm/libfmevent.so:=      REALPATH=../../../lib/fm/libfmevent.so.1
1140 $(ROOT)/usr/lib/fm/llib-lfmevent.ln:=   REALPATH=../../../lib/fm/llib-lfmevent.l
1141 $(ROOT)/usr/lib/fm/llib-lfmevent:=      REALPATH=../../../lib/fm/llib-lfmevent

1143 $(ROOT)/lib/$(MACH64)/libposix4.so.1:= \
1144         REALPATH=librt.so.1
1145 $(ROOT)/lib/$(MACH64)/libposix4.so:= \
1146         REALPATH=libposix4.so.1
1147 $(ROOT)/lib/$(MACH64)/llib-lposix4.ln:= \
1148         REALPATH=llib-lrt.ln
1149 $(ROOT)/lib/$(MACH64)/libthread_db.so.1:= \
1150         REALPATH=libc_db.so.1
1151 $(ROOT)/lib/$(MACH64)/libthread_db.so:= \
1152         REALPATH=libc_db.so.1
1153 $(ROOT)/usr/lib/$(MACH64)/ld.so.1:= \
1154         REALPATH=../../../lib/$(MACH64)/ld.so.1
1155 $(ROOT)/usr/lib/$(MACH64)/libadm.so.1:= \
1156         REALPATH=../../../lib/$(MACH64)/libadm.so.1
1157 $(ROOT)/usr/lib/$(MACH64)/libadm.so:= \
1158         REALPATH=../../../lib/$(MACH64)/libadm.so.1
1159 $(ROOT)/usr/lib/$(MACH64)/libaio.so.1:= \
1160         REALPATH=../../../lib/$(MACH64)/libaio.so.1
1161 $(ROOT)/usr/lib/$(MACH64)/libaio.so:= \
1162         REALPATH=../../../lib/$(MACH64)/libaio.so.1
1163 $(ROOT)/usr/lib/$(MACH64)/libavl.so.1:= \
1164         REALPATH=../../../lib/$(MACH64)/libavl.so.1
1165 $(ROOT)/usr/lib/$(MACH64)/libavl.so:= \
1166         REALPATH=../../../lib/$(MACH64)/libavl.so.1
1167 $(ROOT)/usr/lib/$(MACH64)/libbsm.so.1:= \
1168         REALPATH=../../../lib/$(MACH64)/libbsm.so.1
1169 $(ROOT)/usr/lib/$(MACH64)/libbsm.so:= \
1170         REALPATH=../../../lib/$(MACH64)/libbsm.so.1
1171 $(ROOT)/usr/lib/$(MACH64)/libc.so.1:= \
1172         REALPATH=../../../lib/$(MACH64)/libc.so.1
1173 $(ROOT)/usr/lib/$(MACH64)/libc.so:= \
1174         REALPATH=../../../lib/$(MACH64)/libc.so.1
1175 $(ROOT)/usr/lib/$(MACH64)/libc_db.so.1:= \
1176         REALPATH=../../../lib/$(MACH64)/libc_db.so.1
1177 $(ROOT)/usr/lib/$(MACH64)/libc_db.so:= \
```

```
1178         REALPATH=../../../lib/$(MACH64)/libc_db.so.1
1179 $(ROOT)/usr/lib/$(MACH64)/libcmdutils.so.1:= \
1180         REALPATH=../../../lib/$(MACH64)/libcmdutils.so.1
1181 $(ROOT)/usr/lib/$(MACH64)/libcmdutils.so:= \
1182         REALPATH=../../../lib/$(MACH64)/libcmdutils.so.1
1183 $(ROOT)/usr/lib/$(MACH64)/libcontract.so.1:= \
1184         REALPATH=../../../lib/$(MACH64)/libcontract.so.1
1185 $(ROOT)/usr/lib/$(MACH64)/libcontract.so:= \
1186         REALPATH=../../../lib/$(MACH64)/libcontract.so.1
1187 $(ROOT)/usr/lib/$(MACH64)/libctf.so.1:= \
1188         REALPATH=../../../lib/$(MACH64)/libctf.so.1
1189 $(ROOT)/usr/lib/$(MACH64)/libctf.so:= \
1190         REALPATH=../../../lib/$(MACH64)/libctf.so.1
1191 $(ROOT)/usr/lib/$(MACH64)/libcurses.so.1:= \
1192         REALPATH=../../../lib/$(MACH64)/libcurses.so.1
1193 $(ROOT)/usr/lib/$(MACH64)/libcurses.so:= \
1194         REALPATH=../../../lib/$(MACH64)/libcurses.so.1
1195 $(ROOT)/usr/lib/$(MACH64)/libdevice.so.1:= \
1196         REALPATH=../../../lib/$(MACH64)/libdevice.so.1
1197 $(ROOT)/usr/lib/$(MACH64)/libdevice.so:= \
1198         REALPATH=../../../lib/$(MACH64)/libdevice.so.1
1199 $(ROOT)/usr/lib/$(MACH64)/libdevid.so.1:= \
1200         REALPATH=../../../lib/$(MACH64)/libdevid.so.1
1201 $(ROOT)/usr/lib/$(MACH64)/libdevid.so:= \
1202         REALPATH=../../../lib/$(MACH64)/libdevid.so.1
1203 $(ROOT)/usr/lib/$(MACH64)/libdevinfo.so.1:= \
1204         REALPATH=../../../lib/$(MACH64)/libdevinfo.so.1
1205 $(ROOT)/usr/lib/$(MACH64)/libdevinfo.so:= \
1206         REALPATH=../../../lib/$(MACH64)/libdevinfo.so.1
1207 $(ROOT)/usr/lib/$(MACH64)/libdhcputil.so.1:= \
1208         REALPATH=../../../lib/$(MACH64)/libdhcputil.so.1
1209 $(ROOT)/usr/lib/$(MACH64)/libdhcputil.so:= \
1210         REALPATH=../../../lib/$(MACH64)/libdhcputil.so.1
1211 $(ROOT)/usr/lib/$(MACH64)/libdl.so.1:= \
1212         REALPATH=../../../lib/$(MACH64)/libdl.so.1
1213 $(ROOT)/usr/lib/$(MACH64)/libdl.so:= \
1214         REALPATH=../../../lib/$(MACH64)/libdl.so.1
1215 $(ROOT)/usr/lib/$(MACH64)/libdlpi.so.1:= \
1216         REALPATH=../../../lib/$(MACH64)/libdlpi.so.1
1217 $(ROOT)/usr/lib/$(MACH64)/libdlpi.so:= \
1218         REALPATH=../../../lib/$(MACH64)/libdlpi.so.1
1219 $(ROOT)/usr/lib/$(MACH64)/libdoor.so.1:= \
1220         REALPATH=../../../lib/$(MACH64)/libdoor.so.1
1221 $(ROOT)/usr/lib/$(MACH64)/libdoor.so:= \
1222         REALPATH=../../../lib/$(MACH64)/libdoor.so.1
1223 $(ROOT)/usr/lib/$(MACH64)/libefi.so.1:= \
1224         REALPATH=../../../lib/$(MACH64)/libefi.so.1
1225 $(ROOT)/usr/lib/$(MACH64)/libefi.so:= \
1226         REALPATH=../../../lib/$(MACH64)/libefi.so.1
1227 $(ROOT)/usr/lib/$(MACH64)/libelf.so.1:= \
1228         REALPATH=../../../lib/$(MACH64)/libelf.so.1
1229 $(ROOT)/usr/lib/$(MACH64)/libelf.so:= \
1230         REALPATH=../../../lib/$(MACH64)/libelf.so.1
1231 $(ROOT)/usr/lib/$(MACH64)/libgen.so.1:= \
1232         REALPATH=../../../lib/$(MACH64)/libgen.so.1
1233 $(ROOT)/usr/lib/$(MACH64)/libgen.so:= \
1234         REALPATH=../../../lib/$(MACH64)/libgen.so.1
1235 $(ROOT)/usr/lib/$(MACH64)/libinetutil.so.1:= \
1236         REALPATH=../../../lib/$(MACH64)/libinetutil.so.1
1237 $(ROOT)/usr/lib/$(MACH64)/libinetutil.so:= \
1238         REALPATH=../../../lib/$(MACH64)/libinetutil.so.1
1239 $(ROOT)/usr/lib/$(MACH64)/libintl.so.1:= \
1240         REALPATH=../../../lib/$(MACH64)/libintl.so.1
1241 $(ROOT)/usr/lib/$(MACH64)/libintl.so:= \
1242         REALPATH=../../../lib/$(MACH64)/libintl.so.1
1243 $(ROOT)/usr/lib/$(MACH64)/libkstat.so.1:= \
```



```
1244         REALPATH=../../../lib/$(MACH64)/libkstat.so.1
1245 $(ROOT)/usr/lib/$(MACH64)/libkstat.so:= \
1246         REALPATH=../../../lib/$(MACH64)/libkstat.so.1
1247 $(ROOT)/usr/lib/$(MACH64)/liblddbg.so.4:= \
1248         REALPATH=../../../lib/$(MACH64)/liblddbg.so.4
1249 $(ROOT)/usr/lib/$(MACH64)/libmd.so.1:= \
1250         REALPATH=../../../lib/$(MACH64)/libmd.so.1
1251 $(ROOT)/usr/lib/$(MACH64)/libmd.so:= \
1252         REALPATH=../../../lib/$(MACH64)/libmd.so.1
1253 $(ROOT)/usr/lib/$(MACH64)/libmd5.so.1:= \
1254         REALPATH=../../../lib/$(MACH64)/libmd5.so.1
1255 $(ROOT)/usr/lib/$(MACH64)/libmd5.so:= \
1256         REALPATH=../../../lib/$(MACH64)/libmd5.so.1
1257 $(ROOT)/usr/lib/$(MACH64)/libmp.so.2:= \
1258         REALPATH=../../../lib/$(MACH64)/libmp.so.2
1259 $(ROOT)/usr/lib/$(MACH64)/libmp.so:= \
1260         REALPATH=../../../lib/$(MACH64)/libmp.so.2
1261 $(ROOT)/usr/lib/$(MACH64)/libnsl.so.1:= \
1262         REALPATH=../../../lib/$(MACH64)/libnsl.so.1
1263 $(ROOT)/usr/lib/$(MACH64)/libnsl.so:= \
1264         REALPATH=../../../lib/$(MACH64)/libnsl.so.1
1265 $(ROOT)/usr/lib/$(MACH64)/libnvpair.so.1:= \
1266         REALPATH=../../../lib/$(MACH64)/libnvpair.so.1
1267 $(ROOT)/usr/lib/$(MACH64)/libnvpair.so:= \
1268         REALPATH=../../../lib/$(MACH64)/libnvpair.so.1
1269 $(ROOT)/usr/lib/$(MACH64)/libpam.so.1:= \
1270         REALPATH=../../../lib/$(MACH64)/libpam.so.1
1271 $(ROOT)/usr/lib/$(MACH64)/libpam.so:= \
1272         REALPATH=../../../lib/$(MACH64)/libpam.so.1
1273 $(ROOT)/usr/lib/$(MACH64)/libposix4.so.1:= \
1274         REALPATH=../../../lib/$(MACH64)/librt.so.1
1275 $(ROOT)/usr/lib/$(MACH64)/libposix4.so:= \
1276         REALPATH=../../../lib/$(MACH64)/librt.so.1
1277 $(ROOT)/usr/lib/$(MACH64)/libproc.so.1:= \
1278         REALPATH=../../../lib/$(MACH64)/libproc.so.1
1279 $(ROOT)/usr/lib/$(MACH64)/libproc.so:= \
1280         REALPATH=../../../lib/$(MACH64)/libproc.so.1
1281 $(ROOT)/usr/lib/$(MACH64)/libpthread.so.1:= \
1282         REALPATH=../../../lib/$(MACH64)/libpthread.so.1
1283 $(ROOT)/usr/lib/$(MACH64)/libpthread.so:= \
1284         REALPATH=../../../lib/$(MACH64)/libpthread.so.1
1285 $(ROOT)/usr/lib/$(MACH64)/librcm.so.1:= \
1286         REALPATH=../../../lib/$(MACH64)/librcm.so.1
1287 $(ROOT)/usr/lib/$(MACH64)/librcm.so:= \
1288         REALPATH=../../../lib/$(MACH64)/librcm.so.1
1289 $(ROOT)/usr/lib/$(MACH64)/libresolv.so.2:= \
1290         REALPATH=../../../lib/$(MACH64)/libresolv.so.2
1291 $(ROOT)/usr/lib/$(MACH64)/libresolv.so:= \
1292         REALPATH=../../../lib/$(MACH64)/libresolv.so.2
1293 $(ROOT)/usr/lib/$(MACH64)/librestart.so.1:= \
1294         REALPATH=../../../lib/$(MACH64)/librestart.so.1
1295 $(ROOT)/usr/lib/$(MACH64)/librestart.so:= \
1296         REALPATH=../../../lib/$(MACH64)/librestart.so.1
1297 $(ROOT)/usr/lib/$(MACH64)/librpcsvc.so.1:= \
1298         REALPATH=../../../lib/$(MACH64)/librpcsvc.so.1
1299 $(ROOT)/usr/lib/$(MACH64)/librpcsvc.so:= \
1300         REALPATH=../../../lib/$(MACH64)/librpcsvc.so.1
1301 $(ROOT)/usr/lib/$(MACH64)/librt.so.1:= \
1302         REALPATH=../../../lib/$(MACH64)/librt.so.1
1303 $(ROOT)/usr/lib/$(MACH64)/librt.so:= \
1304         REALPATH=../../../lib/$(MACH64)/librt.so.1
1305 $(ROOT)/usr/lib/$(MACH64)/librtld.so.1:= \
1306         REALPATH=../../../lib/$(MACH64)/librtld.so.1
1307 $(ROOT)/usr/lib/$(MACH64)/librtld_db.so.1:= \
1308         REALPATH=../../../lib/$(MACH64)/librtld_db.so.1
1309 $(ROOT)/usr/lib/$(MACH64)/librtld_db.so:= \
```

```
1310         REALPATH=../../../lib/$(MACH64)/librtld_db.so.1
1311 $(ROOT)/usr/lib/$(MACH64)/libscf.so.1:= \
1312         REALPATH=../../../lib/$(MACH64)/libscf.so.1
1313 $(ROOT)/usr/lib/$(MACH64)/libscf.so:= \
1314         REALPATH=../../../lib/$(MACH64)/libscf.so.1
1315 $(ROOT)/usr/lib/$(MACH64)/libsec.so.1:= \
1316         REALPATH=../../../lib/$(MACH64)/libsec.so.1
1317 $(ROOT)/usr/lib/$(MACH64)/libsec.so:= \
1318         REALPATH=../../../lib/$(MACH64)/libsec.so.1
1319 $(ROOT)/usr/lib/$(MACH64)/libsecdb.so.1:= \
1320         REALPATH=../../../lib/$(MACH64)/libsecdb.so.1
1321 $(ROOT)/usr/lib/$(MACH64)/libsecdb.so:= \
1322         REALPATH=../../../lib/$(MACH64)/libsecdb.so.1
1323 $(ROOT)/usr/lib/$(MACH64)/libsendfile.so.1:= \
1324         REALPATH=../../../lib/$(MACH64)/libsendfile.so.1
1325 $(ROOT)/usr/lib/$(MACH64)/libsendfile.so:= \
1326         REALPATH=../../../lib/$(MACH64)/libsendfile.so.1
1327 $(ROOT)/usr/lib/$(MACH64)/libsocket.so.1:= \
1328         REALPATH=../../../lib/$(MACH64)/libsocket.so.1
1329 $(ROOT)/usr/lib/$(MACH64)/libsocket.so:= \
1330         REALPATH=../../../lib/$(MACH64)/libsocket.so.1
1331 $(ROOT)/usr/lib/$(MACH64)/libsysevent.so.1:= \
1332         REALPATH=../../../lib/$(MACH64)/libsysevent.so.1
1333 $(ROOT)/usr/lib/$(MACH64)/libsysevent.so:= \
1334         REALPATH=../../../lib/$(MACH64)/libsysevent.so.1
1335 $(ROOT)/usr/lib/$(MACH64)/libtermcap.so.1:= \
1336         REALPATH=../../../lib/$(MACH64)/libtermcap.so.1
1337 $(ROOT)/usr/lib/$(MACH64)/libtermcap.so:= \
1338         REALPATH=../../../lib/$(MACH64)/libtermcap.so.1
1339 $(ROOT)/usr/lib/$(MACH64)/libtermlib.so.1:= \
1340         REALPATH=../../../lib/$(MACH64)/libcurses.so.1
1341 $(ROOT)/usr/lib/$(MACH64)/libtermlib.so:= \
1342         REALPATH=../../../lib/$(MACH64)/libcurses.so.1
1343 $(ROOT)/usr/lib/$(MACH64)/libthread.so.1:= \
1344         REALPATH=../../../lib/$(MACH64)/libthread.so.1
1345 $(ROOT)/usr/lib/$(MACH64)/libthread.so:= \
1346         REALPATH=../../../lib/$(MACH64)/libthread.so.1
1347 $(ROOT)/usr/lib/$(MACH64)/libthread_db.so.1:= \
1348         REALPATH=../../../lib/$(MACH64)/libc_db.so.1
1349 $(ROOT)/usr/lib/$(MACH64)/libthread_db.so:= \
1350         REALPATH=../../../lib/$(MACH64)/libc_db.so.1
1351 $(ROOT)/usr/lib/$(MACH64)/libtsnet.so.1:= \
1352         REALPATH=../../../lib/$(MACH64)/libtsnet.so.1
1353 $(ROOT)/usr/lib/$(MACH64)/libtsnet.so:= \
1354         REALPATH=../../../lib/$(MACH64)/libtsnet.so.1
1355 $(ROOT)/usr/lib/$(MACH64)/libtsol.so.2:= \
1356         REALPATH=../../../lib/$(MACH64)/libtsol.so.2
1357 $(ROOT)/usr/lib/$(MACH64)/libtsol.so:= \
1358         REALPATH=../../../lib/$(MACH64)/libtsol.so.2
1359 $(ROOT)/usr/lib/$(MACH64)/libumem.so.1:= \
1360         REALPATH=../../../lib/$(MACH64)/libumem.so.1
1361 $(ROOT)/usr/lib/$(MACH64)/libumem.so:= \
1362         REALPATH=../../../lib/$(MACH64)/libumem.so.1
1363 $(ROOT)/usr/lib/$(MACH64)/libuuid.so.1:= \
1364         REALPATH=../../../lib/$(MACH64)/libuuid.so.1
1365 $(ROOT)/usr/lib/$(MACH64)/libuuid.so:= \
1366         REALPATH=../../../lib/$(MACH64)/libuuid.so.1
1367 $(ROOT)/usr/lib/$(MACH64)/libuutil.so.1:= \
1368         REALPATH=../../../lib/$(MACH64)/libuutil.so.1
1369 $(ROOT)/usr/lib/$(MACH64)/libuutil.so:= \
1370         REALPATH=../../../lib/$(MACH64)/libuutil.so.1
1371 $(ROOT)/usr/lib/$(MACH64)/libw.so.1:= \
1372         REALPATH=../../../lib/$(MACH64)/libw.so.1
1373 $(ROOT)/usr/lib/$(MACH64)/libw.so:= \
1374         REALPATH=../../../lib/$(MACH64)/libw.so.1
1375 $(ROOT)/usr/lib/$(MACH64)/libxnet.so.1:= \
```



```
1376         REALPATH=../../../lib/$(MACH64)/libxnet.so.1
1377 $(ROOT)/usr/lib/$(MACH64)/libxnet.so:= \
1378         REALPATH=../../../lib/$(MACH64)/libxnet.so.1
1379 $(ROOT)/usr/lib/$(MACH64)/libzfs.so:= \
1380         REALPATH=../../../lib/$(MACH64)/libzfs.so.1
1381 $(ROOT)/usr/lib/$(MACH64)/libzfs.so.1:= \
1382         REALPATH=../../../lib/$(MACH64)/libzfs.so.1
1383 $(ROOT)/usr/lib/$(MACH64)/libzfs_core.so:= \
1384         REALPATH=../../../lib/$(MACH64)/libzfs_core.so.1
1385 $(ROOT)/usr/lib/$(MACH64)/libzfs_core.so.1:= \
1386         REALPATH=../../../lib/$(MACH64)/libzfs_core.so.1
1387 $(ROOT)/usr/lib/$(MACH64)/libfdisk.so.1:= \
1388         REALPATH=../../../lib/$(MACH64)/libfdisk.so.1
1389 $(ROOT)/usr/lib/$(MACH64)/libfdisk.so:= \
1390         REALPATH=../../../lib/$(MACH64)/libfdisk.so.1
1391 $(ROOT)/usr/lib/$(MACH64)/llib-ladm.ln:= \
1392         REALPATH=../../../lib/$(MACH64)/llib-ladm.ln
1393 $(ROOT)/usr/lib/$(MACH64)/llib-laio.ln:= \
1394         REALPATH=../../../lib/$(MACH64)/llib-laio.ln
1395 $(ROOT)/usr/lib/$(MACH64)/llib-lavl.ln:= \
1396         REALPATH=../../../lib/$(MACH64)/llib-lavl.ln
1397 $(ROOT)/usr/lib/$(MACH64)/llib-lbsm.ln:= \
1398         REALPATH=../../../lib/$(MACH64)/llib-lbsm.ln
1399 $(ROOT)/usr/lib/$(MACH64)/llib-lc.ln:= \
1400         REALPATH=../../../lib/$(MACH64)/llib-lc.ln
1401 $(ROOT)/usr/lib/$(MACH64)/llib-lcmdutils.ln:= \
1402         REALPATH=../../../lib/$(MACH64)/llib-lcmdutils.ln
1403 $(ROOT)/usr/lib/$(MACH64)/llib-lcontract.ln:= \
1404         REALPATH=../../../lib/$(MACH64)/llib-lcontract.ln
1405 $(ROOT)/usr/lib/$(MACH64)/llib-lctf.ln:= \
1406         REALPATH=../../../lib/$(MACH64)/llib-lctf.ln
1407 $(ROOT)/usr/lib/$(MACH64)/llib-lcurses.ln:= \
1408         REALPATH=../../../lib/$(MACH64)/llib-lcurses.ln
1409 $(ROOT)/usr/lib/$(MACH64)/llib-ldevice.ln:= \
1410         REALPATH=../../../lib/$(MACH64)/llib-ldevice.ln
1411 $(ROOT)/usr/lib/$(MACH64)/llib-ldevid.ln:= \
1412         REALPATH=../../../lib/$(MACH64)/llib-ldevid.ln
1413 $(ROOT)/usr/lib/$(MACH64)/llib-ldevinfo.ln:= \
1414         REALPATH=../../../lib/$(MACH64)/llib-ldevinfo.ln
1415 $(ROOT)/usr/lib/$(MACH64)/llib-ldhcputil.ln:= \
1416         REALPATH=../../../lib/$(MACH64)/llib-ldhcputil.ln
1417 $(ROOT)/usr/lib/$(MACH64)/llib-ldl.ln:= \
1418         REALPATH=../../../lib/$(MACH64)/llib-ldl.ln
1419 $(ROOT)/usr/lib/$(MACH64)/llib-ldoor.ln:= \
1420         REALPATH=../../../lib/$(MACH64)/llib-ldoor.ln
1421 $(ROOT)/usr/lib/$(MACH64)/llib-lefi.ln:= \
1422         REALPATH=../../../lib/$(MACH64)/llib-lefi.ln
1423 $(ROOT)/usr/lib/$(MACH64)/llib-lelf.ln:= \
1424         REALPATH=../../../lib/$(MACH64)/llib-lelf.ln
1425 $(ROOT)/usr/lib/$(MACH64)/llib-lgen.ln:= \
1426         REALPATH=../../../lib/$(MACH64)/llib-lgen.ln
1427 $(ROOT)/usr/lib/$(MACH64)/llib-linetutil.ln:= \
1428         REALPATH=../../../lib/$(MACH64)/llib-linetutil.ln
1429 $(ROOT)/usr/lib/$(MACH64)/llib-lintl.ln:= \
1430         REALPATH=../../../lib/$(MACH64)/llib-lintl.ln
1431 $(ROOT)/usr/lib/$(MACH64)/llib-lkstat.ln:= \
1432         REALPATH=../../../lib/$(MACH64)/llib-lkstat.ln
1433 $(ROOT)/usr/lib/$(MACH64)/llib-lmd5.ln:= \
1434         REALPATH=../../../lib/$(MACH64)/llib-lmd5.ln
1435 $(ROOT)/usr/lib/$(MACH64)/llib-lnsl.ln:= \
1436         REALPATH=../../../lib/$(MACH64)/llib-lnsl.ln
1437 $(ROOT)/usr/lib/$(MACH64)/llib-lnvpair.ln:= \
1438         REALPATH=../../../lib/$(MACH64)/llib-lnvpair.ln
1439 $(ROOT)/usr/lib/$(MACH64)/llib-lpam.ln:= \
1440         REALPATH=../../../lib/$(MACH64)/llib-lpam.ln
1441 $(ROOT)/usr/lib/$(MACH64)/llib-lposix4.ln:= \
```

```
1442         REALPATH=../../../lib/$(MACH64)/llib-lrt.ln
1443 $(ROOT)/usr/lib/$(MACH64)/llib-lpthread.ln:= \
1444         REALPATH=../../../lib/$(MACH64)/llib-lpthread.ln
1445 $(ROOT)/usr/lib/$(MACH64)/llib-lresolv.ln:= \
1446         REALPATH=../../../lib/$(MACH64)/llib-lresolv.ln
1447 $(ROOT)/usr/lib/$(MACH64)/llib-lrpcsvc.ln:= \
1448         REALPATH=../../../lib/$(MACH64)/llib-lrpcsvc.ln
1449 $(ROOT)/usr/lib/$(MACH64)/llib-lrt.ln:= \
1450         REALPATH=../../../lib/$(MACH64)/llib-lrt.ln
1451 $(ROOT)/usr/lib/$(MACH64)/llib-lrtld_db.ln:= \
1452         REALPATH=../../../lib/$(MACH64)/llib-lrtld_db.ln
1453 $(ROOT)/usr/lib/$(MACH64)/llib-lscf.ln:= \
1454         REALPATH=../../../lib/$(MACH64)/llib-lscf.ln
1455 $(ROOT)/usr/lib/$(MACH64)/llib-lsec.ln:= \
1456         REALPATH=../../../lib/$(MACH64)/llib-lsec.ln
1457 $(ROOT)/usr/lib/$(MACH64)/llib-lsecdb.ln:= \
1458         REALPATH=../../../lib/$(MACH64)/llib-lsecdb.ln
1459 $(ROOT)/usr/lib/$(MACH64)/llib-lsendfile.ln:= \
1460         REALPATH=../../../lib/$(MACH64)/llib-lsendfile.ln
1461 $(ROOT)/usr/lib/$(MACH64)/llib-lsocket.ln:= \
1462         REALPATH=../../../lib/$(MACH64)/llib-lsocket.ln
1463 $(ROOT)/usr/lib/$(MACH64)/llib-lsysevent.ln:= \
1464         REALPATH=../../../lib/$(MACH64)/llib-lsysevent.ln
1465 $(ROOT)/usr/lib/$(MACH64)/llib-ltermcap.ln:= \
1466         REALPATH=../../../lib/$(MACH64)/llib-ltermcap.ln
1467 $(ROOT)/usr/lib/$(MACH64)/llib-ltermlib.ln:= \
1468         REALPATH=../../../lib/$(MACH64)/llib-lcurses.ln
1469 $(ROOT)/usr/lib/$(MACH64)/llib-lthread.ln:= \
1470         REALPATH=../../../lib/$(MACH64)/llib-lthread.ln
1471 $(ROOT)/usr/lib/$(MACH64)/llib-lthread_db.ln:= \
1472         REALPATH=../../../lib/$(MACH64)/llib-lc_db.ln
1473 $(ROOT)/usr/lib/$(MACH64)/llib-ltsnet.ln:= \
1474         REALPATH=../../../lib/$(MACH64)/llib-ltsnet.ln
1475 $(ROOT)/usr/lib/$(MACH64)/llib-ltsol.ln:= \
1476         REALPATH=../../../lib/$(MACH64)/llib-ltsol.ln
1477 $(ROOT)/usr/lib/$(MACH64)/llib-lumem.ln:= \
1478         REALPATH=../../../lib/$(MACH64)/llib-lumem.ln
1479 $(ROOT)/usr/lib/$(MACH64)/llib-luuid.ln:= \
1480         REALPATH=../../../lib/$(MACH64)/llib-luuid.ln
1481 $(ROOT)/usr/lib/$(MACH64)/llib-lxnet.ln:= \
1482         REALPATH=../../../lib/$(MACH64)/llib-lxnet.ln
1483 $(ROOT)/usr/lib/$(MACH64)/llib-lzfs.ln:= \
1484         REALPATH=../../../lib/$(MACH64)/llib-lzfs.ln
1485 $(ROOT)/usr/lib/$(MACH64)/llib-lzfs_core.ln:= \
1486         REALPATH=../../../lib/$(MACH64)/llib-lzfs_core.ln
1487 $(ROOT)/usr/lib/$(MACH64)/llib-lfdisk.ln:= \
1488         REALPATH=../../../lib/$(MACH64)/llib-lfdisk.ln
1489 $(ROOT)/usr/lib/$(MACH64)/nss_compat.so.1:= \
1490         REALPATH=../../../lib/$(MACH64)/nss_compat.so.1
1491 $(ROOT)/usr/lib/$(MACH64)/nss_dns.so.1:= \
1492         REALPATH=../../../lib/$(MACH64)/nss_dns.so.1
1493 $(ROOT)/usr/lib/$(MACH64)/nss_files.so.1:= \
1494         REALPATH=../../../lib/$(MACH64)/nss_files.so.1
1495 $(ROOT)/usr/lib/$(MACH64)/nss_nis.so.1:= \
1496         REALPATH=../../../lib/$(MACH64)/nss_nis.so.1
1497 $(ROOT)/usr/lib/$(MACH64)/nss_user.so.1:= \
1498         REALPATH=../../../lib/$(MACH64)/nss_user.so.1
1499 $(ROOT)/usr/lib/fm/$(MACH64)/libfmevent.so.1:= \
1500         REALPATH=../../../../lib/fm/$(MACH64)/libfmevent.so.1
1501 $(ROOT)/usr/lib/fm/$(MACH64)/libfmevent.so:= \
1502         REALPATH=../../../../lib/fm/$(MACH64)/libfmevent.so.1
1503 $(ROOT)/usr/lib/fm/$(MACH64)/llib-lfmevent.ln:= \
1504         REALPATH=../../../../lib/fm/$(MACH64)/llib-lfmevent.ln

1506 i386_SYM.USRLIB= \
1507         /usr/lib/libfdisk.so \
```



```
1508         /usr/lib/libfdisk.so.1 \
1509         /usr/lib/llib-lfdisk \
1510         /usr/lib/llib-lfdisk.ln

1512 SYM.USRLIB= \
1513         $($(MACH)_SYM.USRLIB)   \
1514         /lib/libposix4.so \
1515         /lib/libposix4.so.1 \
1516         /lib/llib-lposix4 \
1517         /lib/llib-lposix4.ln \
1518         /lib/libthread_db.so \
1519         /lib/libthread_db.so.1 \
1520         /usr/lib/ld.so.1 \
1521         /usr/lib/libadm.so \
1522         /usr/lib/libadm.so.1 \
1523         /usr/lib/libaio.so \
1524         /usr/lib/libaio.so.1 \
1525         /usr/lib/libavl.so \
1526         /usr/lib/libavl.so.1 \
1527         /usr/lib/libbsm.so \
1528         /usr/lib/libbsm.so.1 \
1529         /usr/lib/libc.so \
1530         /usr/lib/libc.so.1 \
1531         /usr/lib/libc_db.so \
1532         /usr/lib/libc_db.so.1 \
1533         /usr/lib/libcmdutils.so \
1534         /usr/lib/libcmdutils.so.1 \
1535         /usr/lib/libcontract.so \
1536         /usr/lib/libcontract.so.1 \
1537         /usr/lib/libctf.so \
1538         /usr/lib/libctf.so.1 \
1539         /usr/lib/libcurses.so \
1540         /usr/lib/libcurses.so.1 \
1541         /usr/lib/libdevice.so \
1542         /usr/lib/libdevice.so.1 \
1543         /usr/lib/libdevid.so \
1544         /usr/lib/libdevid.so.1 \
1545         /usr/lib/libdevinfo.so \
1546         /usr/lib/libdevinfo.so.1 \
1547         /usr/lib/libdhcpagent.so \
1548         /usr/lib/libdhcpagent.so.1 \
1549         /usr/lib/libdhcputil.so \
1550         /usr/lib/libdhcputil.so.1 \
1551         /usr/lib/libdl.so \
1552         /usr/lib/libdl.so.1 \
1553         /usr/lib/libdlpi.so \
1554         /usr/lib/libdlpi.so.1 \
1555         /usr/lib/libdoor.so \
1556         /usr/lib/libdoor.so.1 \
1557         /usr/lib/libefi.so \
1558         /usr/lib/libefi.so.1 \
1559         /usr/lib/libelf.so \
1560         /usr/lib/libelf.so.1 \
1561         /usr/lib/libgen.so \
1562         /usr/lib/libgen.so.1 \
1563         /usr/lib/libinetutil.so \
1564         /usr/lib/libinetutil.so.1 \
1565         /usr/lib/libintl.so \
1566         /usr/lib/libintl.so.1 \
1567         /usr/lib/libkstat.so \
1568         /usr/lib/libkstat.so.1 \
1569         /usr/lib/liblddbg.so.4 \
1570         /usr/lib/libmd.so \
1571         /usr/lib/libmd.so.1 \
1572         /usr/lib/libmd5.so \
1573         /usr/lib/libmd5.so.1 \
```

```
1574         /usr/lib/libmeta.so \
1575         /usr/lib/libmeta.so.1 \
1576         /usr/lib/libmp.so \
1577         /usr/lib/libmp.so.1 \
1578         /usr/lib/libmp.so.2 \
1579         /usr/lib/libnsl.so \
1580         /usr/lib/libnsl.so.1 \
1581         /usr/lib/libnvpair.so \
1582         /usr/lib/libnvpair.so.1 \
1583         /usr/lib/libpam.so \
1584         /usr/lib/libpam.so.1 \
1585         /usr/lib/libposix4.so \
1586         /usr/lib/libposix4.so.1 \
1587         /usr/lib/libproc.so \
1588         /usr/lib/libproc.so.1 \
1589         /usr/lib/libpthread.so \
1590         /usr/lib/libpthread.so.1 \
1591         /usr/lib/librcm.so \
1592         /usr/lib/librcm.so.1 \
1593         /usr/lib/libresolv.so \
1594         /usr/lib/libresolv.so.1 \
1595         /usr/lib/libresolv.so.2 \
1596         /usr/lib/librestart.so \
1597         /usr/lib/librestart.so.1 \
1598         /usr/lib/librpcsvc.so \
1599         /usr/lib/librpcsvc.so.1 \
1600         /usr/lib/librt.so \
1601         /usr/lib/librt.so.1 \
1602         /usr/lib/librtld.so.1 \
1603         /usr/lib/librtld_db.so \
1604         /usr/lib/librtld_db.so.1 \
1605         /usr/lib/libscf.so \
1606         /usr/lib/libscf.so.1 \
1607         /usr/lib/libsec.so \
1608         /usr/lib/libsec.so.1 \
1609         /usr/lib/libsecdb.so \
1610         /usr/lib/libsecdb.so.1 \
1611         /usr/lib/libsendfile.so \
1612         /usr/lib/libsendfile.so.1 \
1613         /usr/lib/libsocket.so \
1614         /usr/lib/libsocket.so.1 \
1615         /usr/lib/libsysevent.so \
1616         /usr/lib/libsysevent.so.1 \
1617         /usr/lib/libtermcap.so \
1618         /usr/lib/libtermcap.so.1 \
1619         /usr/lib/libtermlib.so \
1620         /usr/lib/libtermlib.so.1 \
1621         /usr/lib/libthread.so \
1622         /usr/lib/libthread.so.1 \
1623         /usr/lib/libthread_db.so \
1624         /usr/lib/libthread_db.so.1 \
1625         /usr/lib/libtsnet.so \
1626         /usr/lib/libtsnet.so.1 \
1627         /usr/lib/libtsol.so \
1628         /usr/lib/libtsol.so.2 \
1629         /usr/lib/libumem.so \
1630         /usr/lib/libumem.so.1 \
1631         /usr/lib/libuuid.so \
1632         /usr/lib/libuuid.so.1 \
1633         /usr/lib/libuutil.so \
1634         /usr/lib/libuutil.so.1 \
1635         /usr/lib/libw.so \
1636         /usr/lib/libw.so.1 \
1637         /usr/lib/libxnet.so \
1638         /usr/lib/libxnet.so.1 \
1639         /usr/lib/libzfs.so \
```



```
1640         /usr/lib/libzfs.so.1 \
1641         /usr/lib/libzfs_core.so \
1642         /usr/lib/libzfs_core.so.1 \
1643         /usr/lib/llib-ladm \
1644         /usr/lib/llib-ladm.ln \
1645         /usr/lib/llib-laio \
1646         /usr/lib/llib-laio.ln \
1647         /usr/lib/llib-lavl \
1648         /usr/lib/llib-lavl.ln \
1649         /usr/lib/llib-lbsm \
1650         /usr/lib/llib-lbsm.ln \
1651         /usr/lib/llib-lc \
1652         /usr/lib/llib-lc.ln \
1653         /usr/lib/llib-lcmdutils \
1654         /usr/lib/llib-lcmdutils.ln \
1655         /usr/lib/llib-lcontract \
1656         /usr/lib/llib-lcontract.ln \
1657         /usr/lib/llib-lctf \
1658         /usr/lib/llib-lctf.ln \
1659         /usr/lib/llib-lcurses \
1660         /usr/lib/llib-lcurses.ln \
1661         /usr/lib/llib-ldevice \
1662         /usr/lib/llib-ldevice.ln \
1663         /usr/lib/llib-ldevid \
1664         /usr/lib/llib-ldevid.ln \
1665         /usr/lib/llib-ldevinfo \
1666         /usr/lib/llib-ldevinfo.ln \
1667         /usr/lib/llib-ldhcpagent \
1668         /usr/lib/llib-ldhcpagent.ln \
1669         /usr/lib/llib-ldhcputil \
1670         /usr/lib/llib-ldhcputil.ln \
1671         /usr/lib/llib-ldl \
1672         /usr/lib/llib-ldl.ln \
1673         /usr/lib/llib-ldoor \
1674         /usr/lib/llib-ldoor.ln \
1675         /usr/lib/llib-lefi \
1676         /usr/lib/llib-lefi.ln \
1677         /usr/lib/llib-lelf \
1678         /usr/lib/llib-lelf.ln \
1679         /usr/lib/llib-lgen \
1680         /usr/lib/llib-lgen.ln \
1681         /usr/lib/llib-linetutil \
1682         /usr/lib/llib-linetutil.ln \
1683         /usr/lib/llib-lintl \
1684         /usr/lib/llib-lintl.ln \
1685         /usr/lib/llib-lkstat \
1686         /usr/lib/llib-lkstat.ln \
1687         /usr/lib/llib-lmd5 \
1688         /usr/lib/llib-lmd5.ln \
1689         /usr/lib/llib-lmeta \
1690         /usr/lib/llib-lmeta.ln \
1691         /usr/lib/llib-lnsl \
1692         /usr/lib/llib-lnsl.ln \
1693         /usr/lib/llib-lnvpair \
1694         /usr/lib/llib-lnvpair.ln \
1695         /usr/lib/llib-lpam \
1696         /usr/lib/llib-lpam.ln \
1697         /usr/lib/llib-lposix4 \
1698         /usr/lib/llib-lposix4.ln \
1699         /usr/lib/llib-lpthread \
1700         /usr/lib/llib-lpthread.ln \
1701         /usr/lib/llib-lresolv \
1702         /usr/lib/llib-lresolv.ln \
1703         /usr/lib/llib-lrpcsvc \
1704         /usr/lib/llib-lrpcsvc.ln \
1705         /usr/lib/llib-lrt \
```

```
1706         /usr/lib/llib-lrt.ln \
1707         /usr/lib/llib-lrtld_db \
1708         /usr/lib/llib-lrtld_db.ln \
1709         /usr/lib/llib-lscf \
1710         /usr/lib/llib-lscf.ln \
1711         /usr/lib/llib-lsec \
1712         /usr/lib/llib-lsec.ln \
1713         /usr/lib/llib-lsecdb \
1714         /usr/lib/llib-lsecdb.ln \
1715         /usr/lib/llib-lsendfile \
1716         /usr/lib/llib-lsendfile.ln \
1717         /usr/lib/llib-lsocket \
1718         /usr/lib/llib-lsocket.ln \
1719         /usr/lib/llib-lsysevent \
1720         /usr/lib/llib-lsysevent.ln \
1721         /usr/lib/llib-ltermcap \
1722         /usr/lib/llib-ltermcap.ln \
1723         /usr/lib/llib-ltermlib \
1724         /usr/lib/llib-ltermlib.ln \
1725         /usr/lib/llib-lthread \
1726         /usr/lib/llib-lthread.ln \
1727         /usr/lib/llib-lthread_db \
1728         /usr/lib/llib-lthread_db.ln \
1729         /usr/lib/llib-ltsnet \
1730         /usr/lib/llib-ltsnet.ln \
1731         /usr/lib/llib-ltsol \
1732         /usr/lib/llib-ltsol.ln \
1733         /usr/lib/llib-lumem \
1734         /usr/lib/llib-lumem.ln \
1735         /usr/lib/llib-luuid \
1736         /usr/lib/llib-luuid.ln \
1737         /usr/lib/llib-lxnet \
1738         /usr/lib/llib-lxnet.ln \
1739         /usr/lib/llib-lzfs \
1740         /usr/lib/llib-lzfs.ln \
1741         /usr/lib/llib-lzfs_core \
1742         /usr/lib/llib-lzfs_core.ln \
1743         /usr/lib/nss_compat.so.1 \
1744         /usr/lib/nss_dns.so.1 \
1745         /usr/lib/nss_files.so.1 \
1746         /usr/lib/nss_nis.so.1 \
1747         /usr/lib/nss_user.so.1 \
1748         /usr/lib/fm/libfmevent.so \
1749         /usr/lib/fm/libfmevent.so.1 \
1750         /usr/lib/fm/llib-lfmevent \
1751         /usr/lib/fm/llib-lfmevent.ln

1753 sparcv9_SYM.USRLIB64=

1755 amd64_SYM.USRLIB64=     \
1756         /usr/lib/amd64/libfdisk.so      \
1757         /usr/lib/amd64/libfdisk.so.1    \
1758         /usr/lib/amd64/llib-lfdisk.ln


1761 SYM.USRLIB64= \
1762         $($(MACH64)_SYM.USRLIB64)               \
1763         /lib/$(MACH64)/libposix4.so \
1764         /lib/$(MACH64)/libposix4.so.1 \
1765         /lib/$(MACH64)/llib-lposix4.ln \
1766         /lib/$(MACH64)/libthread_db.so \
1767         /lib/$(MACH64)/libthread_db.so.1 \
1768         /usr/lib/$(MACH64)/ld.so.1 \
1769         /usr/lib/$(MACH64)/libadm.so \
1770         /usr/lib/$(MACH64)/libadm.so.1 \
1771         /usr/lib/$(MACH64)/libaio.so \
```



```
1772         /usr/lib/$(MACH64)/libaio.so.1 \
1773         /usr/lib/$(MACH64)/libavl.so \
1774         /usr/lib/$(MACH64)/libavl.so.1 \
1775         /usr/lib/$(MACH64)/libbsm.so \
1776         /usr/lib/$(MACH64)/libbsm.so.1 \
1777         /usr/lib/$(MACH64)/libc.so \
1778         /usr/lib/$(MACH64)/libc.so.1 \
1779         /usr/lib/$(MACH64)/libc_db.so \
1780         /usr/lib/$(MACH64)/libc_db.so.1 \
1781         /usr/lib/$(MACH64)/libcmdutils.so \
1782         /usr/lib/$(MACH64)/libcmdutils.so.1 \
1783         /usr/lib/$(MACH64)/libcontract.so \
1784         /usr/lib/$(MACH64)/libcontract.so.1 \
1785         /usr/lib/$(MACH64)/libctf.so \
1786         /usr/lib/$(MACH64)/libctf.so.1 \
1787         /usr/lib/$(MACH64)/libcurses.so \
1788         /usr/lib/$(MACH64)/libcurses.so.1 \
1789         /usr/lib/$(MACH64)/libdevice.so \
1790         /usr/lib/$(MACH64)/libdevice.so.1 \
1791         /usr/lib/$(MACH64)/libdevid.so \
1792         /usr/lib/$(MACH64)/libdevid.so.1 \
1793         /usr/lib/$(MACH64)/libdevinfo.so \
1794         /usr/lib/$(MACH64)/libdevinfo.so.1 \
1795         /usr/lib/$(MACH64)/libdhcputil.so \
1796         /usr/lib/$(MACH64)/libdhcputil.so.1 \
1797         /usr/lib/$(MACH64)/libdl.so \
1798         /usr/lib/$(MACH64)/libdl.so.1 \
1799         /usr/lib/$(MACH64)/libdlpi.so \
1800         /usr/lib/$(MACH64)/libdlpi.so.1 \
1801         /usr/lib/$(MACH64)/libdoor.so \
1802         /usr/lib/$(MACH64)/libdoor.so.1 \
1803         /usr/lib/$(MACH64)/libefi.so \
1804         /usr/lib/$(MACH64)/libefi.so.1 \
1805         /usr/lib/$(MACH64)/libelf.so \
1806         /usr/lib/$(MACH64)/libelf.so.1 \
1807         /usr/lib/$(MACH64)/libgen.so \
1808         /usr/lib/$(MACH64)/libgen.so.1 \
1809         /usr/lib/$(MACH64)/libinetutil.so \
1810         /usr/lib/$(MACH64)/libinetutil.so.1 \
1811         /usr/lib/$(MACH64)/libintl.so \
1812         /usr/lib/$(MACH64)/libintl.so.1 \
1813         /usr/lib/$(MACH64)/libkstat.so \
1814         /usr/lib/$(MACH64)/libkstat.so.1 \
1815         /usr/lib/$(MACH64)/liblddbg.so.4 \
1816         /usr/lib/$(MACH64)/libmd.so \
1817         /usr/lib/$(MACH64)/libmd.so.1 \
1818         /usr/lib/$(MACH64)/libmd5.so \
1819         /usr/lib/$(MACH64)/libmd5.so.1 \
1820         /usr/lib/$(MACH64)/libmp.so \
1821         /usr/lib/$(MACH64)/libmp.so.2 \
1822         /usr/lib/$(MACH64)/libnsl.so \
1823         /usr/lib/$(MACH64)/libnsl.so.1 \
1824         /usr/lib/$(MACH64)/libnvpair.so \
1825         /usr/lib/$(MACH64)/libnvpair.so.1 \
1826         /usr/lib/$(MACH64)/libpam.so \
1827         /usr/lib/$(MACH64)/libpam.so.1 \
1828         /usr/lib/$(MACH64)/libposix4.so \
1829         /usr/lib/$(MACH64)/libposix4.so.1 \
1830         /usr/lib/$(MACH64)/libproc.so \
1831         /usr/lib/$(MACH64)/libproc.so.1 \
1832         /usr/lib/$(MACH64)/libpthread.so \
1833         /usr/lib/$(MACH64)/libpthread.so.1 \
1834         /usr/lib/$(MACH64)/librcm.so \
1835         /usr/lib/$(MACH64)/librcm.so.1 \
1836         /usr/lib/$(MACH64)/libresolv.so \
1837         /usr/lib/$(MACH64)/libresolv.so.2 \
```

```
1838         /usr/lib/$(MACH64)/librestart.so \
1839         /usr/lib/$(MACH64)/librestart.so.1 \
1840         /usr/lib/$(MACH64)/librpcsvc.so \
1841         /usr/lib/$(MACH64)/librpcsvc.so.1 \
1842         /usr/lib/$(MACH64)/librt.so \
1843         /usr/lib/$(MACH64)/librt.so.1 \
1844         /usr/lib/$(MACH64)/librtld.so.1 \
1845         /usr/lib/$(MACH64)/librtld_db.so \
1846         /usr/lib/$(MACH64)/librtld_db.so.1 \
1847         /usr/lib/$(MACH64)/libscf.so \
1848         /usr/lib/$(MACH64)/libscf.so.1 \
1849         /usr/lib/$(MACH64)/libsec.so \
1850         /usr/lib/$(MACH64)/libsec.so.1 \
1851         /usr/lib/$(MACH64)/libsecdb.so \
1852         /usr/lib/$(MACH64)/libsecdb.so.1 \
1853         /usr/lib/$(MACH64)/libsendfile.so \
1854         /usr/lib/$(MACH64)/libsendfile.so.1 \
1855         /usr/lib/$(MACH64)/libsocket.so \
1856         /usr/lib/$(MACH64)/libsocket.so.1 \
1857         /usr/lib/$(MACH64)/libsysevent.so \
1858         /usr/lib/$(MACH64)/libsysevent.so.1 \
1859         /usr/lib/$(MACH64)/libtermcap.so \
1860         /usr/lib/$(MACH64)/libtermcap.so.1 \
1861         /usr/lib/$(MACH64)/libtermlib.so \
1862         /usr/lib/$(MACH64)/libtermlib.so.1 \
1863         /usr/lib/$(MACH64)/libthread.so \
1864         /usr/lib/$(MACH64)/libthread.so.1 \
1865         /usr/lib/$(MACH64)/libthread_db.so \
1866         /usr/lib/$(MACH64)/libthread_db.so.1 \
1867         /usr/lib/$(MACH64)/libtsnet.so \
1868         /usr/lib/$(MACH64)/libtsnet.so.1 \
1869         /usr/lib/$(MACH64)/libtsol.so \
1870         /usr/lib/$(MACH64)/libtsol.so.2 \
1871         /usr/lib/$(MACH64)/libumem.so \
1872         /usr/lib/$(MACH64)/libumem.so.1 \
1873         /usr/lib/$(MACH64)/libuuid.so \
1874         /usr/lib/$(MACH64)/libuuid.so.1 \
1875         /usr/lib/$(MACH64)/libuutil.so \
1876         /usr/lib/$(MACH64)/libuutil.so.1 \
1877         /usr/lib/$(MACH64)/libw.so \
1878         /usr/lib/$(MACH64)/libw.so.1 \
1879         /usr/lib/$(MACH64)/libxnet.so \
1880         /usr/lib/$(MACH64)/libxnet.so.1 \
1881         /usr/lib/$(MACH64)/libzfs.so \
1882         /usr/lib/$(MACH64)/libzfs.so.1 \
1883         /usr/lib/$(MACH64)/libzfs_core.so \
1884         /usr/lib/$(MACH64)/libzfs_core.so.1 \
1885         /usr/lib/$(MACH64)/llib-ladm.ln \
1886         /usr/lib/$(MACH64)/llib-laio.ln \
1887         /usr/lib/$(MACH64)/llib-lavl.ln \
1888         /usr/lib/$(MACH64)/llib-lbsm.ln \
1889         /usr/lib/$(MACH64)/llib-lc.ln \
1890         /usr/lib/$(MACH64)/llib-lcmdutils.ln \
1891         /usr/lib/$(MACH64)/llib-lcontract.ln \
1892         /usr/lib/$(MACH64)/llib-lctf.ln \
1893         /usr/lib/$(MACH64)/llib-lcurses.ln \
1894         /usr/lib/$(MACH64)/llib-ldevice.ln \
1895         /usr/lib/$(MACH64)/llib-ldevid.ln \
1896         /usr/lib/$(MACH64)/llib-ldevinfo.ln \
1897         /usr/lib/$(MACH64)/llib-ldhcputil.ln \
1898         /usr/lib/$(MACH64)/llib-ldl.ln \
1899         /usr/lib/$(MACH64)/llib-ldoor.ln \
1900         /usr/lib/$(MACH64)/llib-lefi.ln \
1901         /usr/lib/$(MACH64)/llib-lelf.ln \
1902         /usr/lib/$(MACH64)/llib-lgen.ln \
1903         /usr/lib/$(MACH64)/llib-linetutil.ln \
```



```
1904         /usr/lib/$(MACH64)/llib-lintl.ln \
1905         /usr/lib/$(MACH64)/llib-lkstat.ln \
1906         /usr/lib/$(MACH64)/llib-lmd5.ln \
1907         /usr/lib/$(MACH64)/llib-lnsl.ln \
1908         /usr/lib/$(MACH64)/llib-lnvpair.ln \
1909         /usr/lib/$(MACH64)/llib-lpam.ln \
1910         /usr/lib/$(MACH64)/llib-lposix4.ln \
1911         /usr/lib/$(MACH64)/llib-lpthread.ln \
1912         /usr/lib/$(MACH64)/llib-lresolv.ln \
1913         /usr/lib/$(MACH64)/llib-lrpcsvc.ln \
1914         /usr/lib/$(MACH64)/llib-lrt.ln \
1915         /usr/lib/$(MACH64)/llib-lrtld_db.ln \
1916         /usr/lib/$(MACH64)/llib-lscf.ln \
1917         /usr/lib/$(MACH64)/llib-lsec.ln \
1918         /usr/lib/$(MACH64)/llib-lsecdb.ln \
1919         /usr/lib/$(MACH64)/llib-lsendfile.ln \
1920         /usr/lib/$(MACH64)/llib-lsocket.ln \
1921         /usr/lib/$(MACH64)/llib-lsysevent.ln \
1922         /usr/lib/$(MACH64)/llib-ltermcap.ln \
1923         /usr/lib/$(MACH64)/llib-ltermlib.ln \
1924         /usr/lib/$(MACH64)/llib-lthread.ln \
1925         /usr/lib/$(MACH64)/llib-lthread_db.ln \
1926         /usr/lib/$(MACH64)/llib-ltsnet.ln \
1927         /usr/lib/$(MACH64)/llib-ltsol.ln \
1928         /usr/lib/$(MACH64)/llib-lumem.ln \
1929         /usr/lib/$(MACH64)/llib-luuid.ln \
1930         /usr/lib/$(MACH64)/llib-lxnet.ln \
1931         /usr/lib/$(MACH64)/llib-lzfs.ln \
1932         /usr/lib/$(MACH64)/llib-lzfs_core.ln \
1933         /usr/lib/$(MACH64)/nss_compat.so.1 \
1934         /usr/lib/$(MACH64)/nss_dns.so.1 \
1935         /usr/lib/$(MACH64)/nss_files.so.1 \
1936         /usr/lib/$(MACH64)/nss_nis.so.1 \
1937         /usr/lib/$(MACH64)/nss_user.so.1 \
1938         /usr/lib/fm/$(MACH64)/libfmevent.so \
1939         /usr/lib/fm/$(MACH64)/libfmevent.so.1 \
1940         /usr/lib/fm/$(MACH64)/llib-lfmevent.ln

1942 #
1943 # usr/src/Makefile uses INS.dir for any member of ROOTDIRS, the fact
1944 # these are symlinks to files has no bearing on this.
1945 #
1946 $(FILELINKS:%=$(ROOT)%):= \
1947     INS.dir= -$(RM) $@; $(SYMLINK) $(REALPATH) $@
```

```
**********************************************************
    8596 Thu Jul 11 01:28:50 2013
new/usr/src/cmd/cmd-inet/usr.sbin/Makefile
first pass
**********************************************************
#
# 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) 1990, 2010, Oracle and/or its affiliates. All rights reserved.
#

SYNCPROG=       syncinit syncloop syncstat
DHCPPROG=       dhcpconfig dhtadm pntadm

# EXPORT DELETE START
XMODPROG=       wanbootutil
# EXPORT DELETE END

PROG=           6to4relay arp gettable if_mpadm \
                in.comsat in.fingerd in.rarpd in.rexecd in.rlogind \
                in.rshd in.rwhod in.telnetd in.tftpd ipaddrsel \
                ndd $(SYNCPROG) $(DHCPPROG) wanbootutil
                ndd $(SYNCPROG) $(DHCPPROG) $(XMODPROG)

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

ROOTFS_PROG=    hostconfig route soconfig
SBINLINKS=      hostconfig route

RPCSVCPROG=     hostconfig
AUDITPROG=      in.rexecd in.rlogind in.rshd in.telnetd
PAMPROG=        in.rexecd in.rlogind in.rshd in.telnetd
SOCKETPROG=     6to4relay arp gettable hostconfig if_mpadm in.comsat \
                in.fingerd in.rarpd in.rexecd in.rlogind in.rshd \
                in.rwhod in.telnetd in.tftpd ipaddrsel route
NSLPROG=        6to4relay arp gettable hostconfig in.comsat in.rarpd \
                in.rexecd in.rlogind in.rshd in.rwhod in.telnetd \
                in.tftpd ipaddrsel route
CMDPROG=        in.telnetd
K5PROGS=        in.telnetd in.rlogind in.rshd
TSNETPROG=      route
DLADMPROG=      6to4relay
DEFAULTFILES=   telnetd.dfl

PROGSRCS=       $(PROG:%=%.c)
```



```
TFTPDOBJS=      in.tftpd.o tftpsubs.o
OTHERSRC=       ../usr.bin/tftp/tftpsubs.c
K5RLOGINOBJS=   in.rlogind.o
K5RSHDOBJS=     in.rshd.o
K5TELNETOBJS=   in.telnetd.o
SRCS=           $(PROGSRCS) $(OTHERSRC)

SUBDIRS=        bootconfchk htable ifconfig ilbadm in.ftpd in.rdisc in.routed \
                in.talkd inetadm inetconv ipadm ipmpstat ipqosconf ipsecutils \
                kssl/kssladm kssl/ksslcfg nwamadm nwamcfg ping routeadm \
                snoop sppptun traceroute wificonfig

MSGSUBDIRS=     bootconfchk htable ifconfig ilbadm in.ftpd in.routed in.talkd \
                inetadm inetconv ipadm ipmpstat ipqosconf ipsecutils \
                kssl/ksslcfg nwamadm nwamcfg routeadm sppptun snoop wificonfig

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

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

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

COMMONOBJS=     addr_match.o kcmd.o store_forw_creds.o
COMMONSRCS=     $(COMMONOBJS:%.o=$(CMDINETCOMMONDIR)/%.c)
SRCS+=          $(COMMONSRCS)

CERRWARN += -_gcc=-Wno-implicit-function-declaration
CERRWARN += -_gcc=-Wno-uninitialized
CERRWARN += -_gcc=-Wno-unused-variable
CERRWARN += -_gcc=-Wno-unused-function
CERRWARN += -_gcc=-Wno-parentheses
CERRWARN += -_gcc=-Wno-char-subscripts
CERRWARN += -_gcc=-Wno-extra
CERRWARN += -_gcc=-Wno-address

#
# Message catalog
#
POFILES=        6to4relay.po if_mpadm.po in.comsat.po ipaddrsel.po route.po \
                soconfig.po
POFILE=         usr.sbin.po

all:=           TARGET= all
install:=       TARGET= install
clean:=         TARGET= clean
clobber:=       TARGET= clobber
```

```
123 lint:=          TARGET= lint
124 _msg:=          TARGET= _msg

126 CLOBBERFILES += $(ROOTFS_PROG) $(PROG)
127 CLEANFILES += $(COMMONOBJS) $(K5RLOGINOBJS) $(K5RSHDOBJS) $(TFTPDOBJS)

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

131 include $(SRC)/lib/gss_mechs/mech_krb5/Makefile.mech_krb5
132 K5LIBS=

134 # Eventually just plain CFLAGS should be += -v, but not until all in
135 # PROGS are lint clean.
136 $(LINTCLEAN)            :=      CFLAGS += $(CCVERBOSE)
137 $(CONSTCLEAN)           :=      CFLAGS += $(XSTRCONST)

139 $(SYNCPROG)             :=      LDLIBS += -ldlpi
140 $(SOCKETPROG)           :=      LDLIBS += -lsocket
141 $(NSLPROG)              :=      LDLIBS += -lnsl
142 $(AUDITPROG)            :=      LDLIBS += -lbsm
143 $(PAMPROG)              :=      LDLIBS += -lpam
144 $(RPCSVCPROG)           :=      LDLIBS += -lrpcsvc
145 $(K5PROGS)              :=      LDFLAGS += $(KRUNPATH) \
146                                 -L$(ROOT)$(KLIBDIR_DO) -L$(ROOT)$(KLIBDIR_GL)
147 $(K5PROGS)              :=      K5LIBS=  -lmech_krb5
148 $(K5PROGS)              :=      CPPFLAGS += -I$(SRC)/head \
149                                 -I$(SRC)/uts/common/ \
150                                 -I$(SRC)/uts/common/gssapi/mechs/krb5/include \
151                                 -I$(SRC)/lib/gss_mechs/mech_krb5/include \
152                                 -I$(SRC)/lib/pam_modules/krb5
153 LDLIBS +=       $(K5LIBS)
154 $(TSNETPROG)            :=      LDLIBS += -ltsnet
155 $(DLADMPROG)            :=      LDLIBS += -ldladm

157 in.rarpd                :=      LDLIBS += -linetutil -ldlpi
158 if_mpadm                :=      LDLIBS += -linetutil -lipmp
159 if_mpadm.po             :=      XGETFLAGS += -a
160 route                   :=      CPPFLAGS += -DNDEBUG
161 ndd                     :=      LDLIBS += -ldladm -lipadm
162 gettable in.comsat      :=      LDFLAGS += $(MAPFILE.NGB:%=-M%)

164 .KEEP_STATE:

166 .PARALLEL:

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

170 #
171 # message catalog
172 #
173 _msg: $(MSGSUBDIRS) $(POFILE)

175 syncutil: $(SYNCPROG)

177 $(POFILE): $(POFILES)
178         $(RM) $@
179         cat $(POFILES) > $@

181 %.o: $(CMDINETCOMMONDIR)/%.c
182         $(COMPILE.c) -o $@ $<

184 in.telnetd: $(K5TELNETOBJS)
185         $(LINK.c) $(K5TELNETOBJS) -o $@ $(LDLIBS)
186         $(POST_PROCESS)

188 in.rlogind: $(K5RLOGINOBJS) $(COMMONOBJS)
```



```
189         $(LINK.c) $(K5RLOGINOBJS) $(COMMONOBJS) -o $@ $(LDLIBS)
190         $(POST_PROCESS)

192 in.rshd: $(K5RSHDOBJS) $(COMMONOBJS)
193         $(LINK.c) $(K5RSHDOBJS) $(COMMONOBJS) -o $@ $(LDLIBS)
194         $(POST_PROCESS)

196 in.tftpd: $(TFTPDOBJS)
197         $(LINK.c) $(TFTPDOBJS) -o $@ $(LDLIBS)
198         $(POST_PROCESS)

200 tftpsubs.o: $(OTHERSRC)
201         $(COMPILE.c) $(OTHERSRC) -o $@
202         $(POST_PROCESS_O)

204 $(ROOTUSRSBINLINKS):
205         -$(RM) $@; $(SYMLINK) ../../sbin/$(@F) $@

207 install: $(PROG) $(ROOTFS_PROG) $(SUBDIRS) .WAIT $(ROOTUSRSBINPROG) \
208         $(ROOTSBINPROG) $(ROOTUSRSBINLINKS) $(ROOTETCDEFAULTFILES) \
209         $(ROOTMANIFEST) $(ROOTSVCMETHOD) THIRDPARTYLICENSE.arp

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

214 CLOBBERFILES += THIRDPARTYLICENSE.arp

216 #
217 # The reason this rule checks for the existence of the
218 # Makefile is that some of the directories do not exist
219 # in our exportable source builds.
220 #
221 $(SUBDIRS): FRC
222         @if [ -f $@/Makefile  ]; then \
223                 cd $@; pwd; $(MAKE) $(TARGET); \
224         else \
225                 true; \
226         fi

228 FRC:

230 check:  $(CHKMANIFEST)

232 clean:  $(SUBDIRS)
233         -$(RM) $(CLEANFILES)

235 clobber: $(SUBDIRS)
236         -$(RM) $(CLEANFILES) $(CLOBBERFILES)

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

```
255                 -erroff=E_GLOBAL_COULD_BE_STATIC2 \
256                 -I$(SRC)/head -I$(SRC)/uts/common/ \
257                 -I$(SRC)/uts/common/gssapi/mechs/krb5/include \
258                 -I$(SRC)/lib/gss_mechs/mech_krb5/include \
259                 -I$(SRC)/lib/pam_modules/krb5 \
260                 in.telnetd.c $(LDLIBS) -lbsm -lpam -lsocket -lnsl
261         $(LINT.c) if_mpadm.c $(LDLIBS) -lsocket -lnsl -lipmp -linetutil
262         $(LINT.c) ipaddrsel.c $(LDLIBS) -lsocket -lnsl
263         $(LINT.c) route.c $(LDLIBS) -lsocket -lnsl -ltsnet
264         $(LINT.c) syncinit.c $(LDLIBS) -ldlpi
265         $(LINT.c) syncloop.c $(LDLIBS) -ldlpi
266         $(LINT.c) syncstat.c $(LDLIBS) -ldlpi
267         $(LINT.c) -erroff=E_NAME_USED_NOT_DEF2 in.rarpd.c $(LDLIBS) \
268             -lsocket -lnsl
269         $(LINT.c) in.tftpd.c ../usr.bin/tftp/tftpsubs.c $(LDLIBS) \
270             -lsocket -lnsl
```

```
276 # EXPORT DELETE START
277 EXPORT_SRC:
278         $(RM) Makefile+
279         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
280                 < Makefile > Makefile+
281         $(RM) Makefile
282         $(MV) Makefile+ Makefile
283         $(CHMOD) 444 Makefile
284 # EXPORT DELETE END
```

```
**********************************************************
    1120 Thu Jul 11 01:28:51 2013
new/usr/src/cmd/crypt/Makefile
first pass
**********************************************************
```

```
   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 #pragma ident   "%Z%%M% %I%     %E% SMI"
  24 #
  25 # Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
  26 # Use is subject to license terms.
  27 #

  29 PROG= crypt

  31 include ../Makefile.cmd

  33 LDLIBS += -lcrypt

  35 .KEEP_STATE:

  37 all: $(PROG)

  39 install: all $(ROOTPROG)

  41 clean:

  43 lint:   lint_PROG

  45 include ../Makefile.targ

  47 # EXPORT DELETE START
  48 EXPORT_SRC:
  49         $(RM) $(PROG).c+ Makefile+
  50         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  51                 < $(PROG).c > $(PROG).c+
  52         $(MV) $(PROG).c+ $(PROG).c
  53         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
  54                 < Makefile > Makefile+
  55         $(RM) Makefile
  56         $(MV) Makefile+ Makefile
  57         $(CHMOD) 444 Makefile $(PROG).c

  59 # EXPORT DELETE END
```

```
**********************************************************
    4307 Thu Jul 11 01:28:51 2013
new/usr/src/cmd/crypt/crypt.c
first pass
**********************************************************
```

```
/*
 * 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 (c) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */


/*
 * Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 *      A one-rotor machine designed along the lines of Enigma
 *      but considerably trivialized.
 */

/* EXPORT DELETE START */
#define ECHO 010
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <crypt.h>
#include <errno.h>

#define ROTORSZ 256
#define MASK 0377
char    t1[ROTORSZ];
char    t2[ROTORSZ];
char    t3[ROTORSZ];

static void
setup(pw)
char *pw;
{
        int ic, i, k, temp;
        unsigned random;
        char buf[13];
        long seed;
        char *ret;
        int err;
```



```
        (void) strncpy(buf, pw, 8);
        buf[8] = buf[0];
        buf[9] = buf[1];
        errno = 0;
        ret = des_crypt(buf, &buf[8]);
        if (ret == NULL) {
                err = errno;
                (void) fprintf(stderr, "crypt: setup failed, unable to"
                    " initialize rotors: %s\n", strerror(err));
                exit(1);
        }
        (void) strncpy(buf, ret, 13);
        seed = 123;
        for (i = 0; i < 13; i++)
                seed = seed*buf[i] + i;
        for (i = 0; i < ROTORSZ; i++) {
                t1[i] = i;
                t3[i] = 0;
        }
        for (i = 0; i < ROTORSZ; i++) {
                seed = 5*seed + buf[i%13];
                random = seed % 65521;
                k = ROTORSZ-1 - i;
                ic = (random&MASK)%(k+1);
                random >>= 8;
                temp = t1[k];
                t1[k] = t1[ic];
                t1[ic] = temp;
                if (t3[k] != 0) continue;
                ic = (random&MASK) % k;
                while (t3[ic] != 0) ic = (ic+1) % k;
                t3[k] = ic;
                t3[ic] = k;
        }
        for (i = 0; i < ROTORSZ; i++)
                t2[t1[i]&MASK] = i;
}
/* EXPORT DELETE END */

int
main(int argc, char **argv)
{
/* EXPORT DELETE START */
        extern int optind;
        char *p1;
        int i, n1, n2, nchar;
        int c;
        struct {
                long offset;
                unsigned int count;
        } header;
        int pflag = 0;
        int kflag = 0;
        char *buf;
        char key[8];
        char keyvar[] = "CrYpTkEy=XXXXXXXX";
        char *s;

        if (argc < 2) {
                if ((buf = (char *)getpass("Enter key:")) == NULL) {
                        (void) fprintf(stderr, "Cannot open /dev/tty\n");
                        exit(1);
                }
                setup(buf);
        } else {
                while ((c = getopt(argc, argv, "pk")) != EOF)
```

```
125                         switch (c) {
126                         case 'p':
127                         /* notify editor that exec has succeeded */
128                                 if (write(1, "y", 1) != 1)
129                                         exit(1);
130                                 if (read(0, key, 8) != 8)
131                                         exit(1);
132                                 setup(key);
133                                 pflag = 1;
134                                 break;
135                         case 'k':
136                                 if ((s = getenv("CrYpTkEy")) == (char *)NULL) {
137                                         (void) fprintf(stderr,
138                                             "CrYpTkEy not set.\n");
139                                         exit(1);
140                                 }
141                                 (void) strncpy(key, s, 8);
142                                 setup(key);
143                                 kflag = 1;
144                                 break;
145                         case '?':
146                                 (void) fprintf(stderr,
147                                     "usage: crypt [ -k ] [ key]\n");
148                                 exit(2);
149                         }
150                 if (pflag == 0 && kflag == 0) {
151                         (void) strncpy(keyvar+9, argv[optind], 8);
152                         (void) putenv(keyvar);
153                         (void) execlp("crypt", "crypt", "-k", 0);
154                 }
155         }
156         if (pflag)
157                 for (;;) {
158                         if ((nchar = read(0, (char *)&header, sizeof (header)))
159                             != sizeof (header))
160                                 exit(nchar);
161                         n1 = (int)(header.offset&MASK);
162                         n2 = (int)((header.offset >> 8) &MASK);
163                         nchar = header.count;
164                         buf = (char *)malloc(nchar);
165                         p1 = buf;
166                         if (read(0, buf, nchar) != nchar)
167                                 exit(1);
168                         while (nchar--) {
169                                 *p1 = t2[(t3[(t1[(*p1 + n1)&MASK]+
170                                     n2)&MASK] - n2)&MASK] - n1;
171                                 n1++;
172                                 if (n1 == ROTORSZ) {
173                                         n1 = 0;
174                                         n2++;
175                                         if (n2 == ROTORSZ) n2 = 0;
176                                 }
177                                 p1++;
178                         }
179                         nchar = header.count;
180                         if (write(1, buf, nchar) != nchar)
181                                 exit(1);
182                         free(buf);
183                 }

185         n1 = 0;
186         n2 = 0;

188         while ((i = getchar()) >= 0) {
189                 i = t2[(t3[(t1[(i+n1)&MASK]+n2)&MASK]-n2)&MASK]-n1;
190                 (void) putchar(i);
```



```
191                 n1++;
192                 if (n1 == ROTORSZ) {
193                         n1 = 0;
194                         n2++;
195                         if (n2 == ROTORSZ) n2 = 0;
196                 }
197         }
198         return (0);
202 /* EXPORT DELETE END */
199 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3535 Thu Jul 11 01:28:52 2013
new/usr/src/cmd/gss/gssd/Makefile
first pass
**********************************************************
```

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

TESTPROG = gssdtest

OUTPUT_OPTION = -I.

PROG= gssd

MANIFEST=       gss.xml

GSSD_BASEOBJS = gssd.o gssd_proc.o gssd_generic.o gssd_getuid.o
GSSC_BASEOBJS = gssdtest.o gssd_release_name_and_type.o gssd_clnt_stubs.o \
                gssd_handle.o

GD_OBJS = gssd_svc.o
GC_OBJS = gssd_clnt.o
G_OBJS  = gssd_xdr.o
GSSDOBJS = $(GSSD_BASEOBJS) $(GD_OBJS) $(G_OBJS)
GSSCOBJS = $(GSSC_BASEOBJS) $(GC_OBJS) $(G_OBJS)

GSSD_LINTS = $(GSSD_BASEOBJS:.o=.c)
GSSC_LINTS = $(GSSC_BASEOBJS:.o=.c)

ROBJS   = $(GD_OBJS) $(GC_OBJS) $(G_OBJS)
OBJS    = $(GSSD_BASEOBJS) $(GD_OBJS) $(GSSC_BASEOBJS) $(GC_OBJS) $(G_OBJS)
SRCS    = $(OBJS:.o=.c)
RSRC    = $(ROBJS:.o=.c)
RSRC    += gssd.h

CLOBBERFILES += $(TESTPROG)

include ../../Makefile.cmd

ROOTMANIFESTDIR=        $(ROOTSVCNETWORKRPC)

TEXT_DOMAIN = SUNW_OST_NETRPC
POFILE = $(PROG).po
POFILES = generic.po
```



```
#
# Override $ROOTLIB
#
ROOTLIB=        $(ROOT)/usr/lib/gss

DIRS=   $(ROOTLIB)

CPPFLAGS += -I$(SRC)/uts/common/gssapi/include
COPTFLAG += $(XESS) #-I$(KINCDIR)

CERRWARN += -_gcc=-Wno-unused-variable
CERRWARN += -_gcc=-Wno-implicit-function-declaration
CERRWARN += -_gcc=-Wno-parentheses
CERRWARN += -_gcc=-Wno-uninitialized

LDLIBS += -lgss -lnsl

gssd := MAPFILES = $(MAPFILE.INT) $(MAPFILE.NGB)
gssd := LDFLAGS += $(MAPFILES:%=-M%)

$(GPROGS) := CPPFLAGS += -DSYSV -DSunOS=50

.KEEP_STATE:

all: $(PROG) $(TESTPROG)

$(ROOTLIB):
        $(INS.dir)

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

gssd:   $(GSSDOBJS) $$(MAPFILES)
        $(LINK.c) $(GSSDOBJS) -o $@ $(LDLIBS)
        $(POST_PROCESS)

gssdtest:       $(GSSCOBJS)
        $(LINK.c) $(GSSCOBJS) -o $@ $(LDLIBS)
        $(POST_PROCESS)

GSSDX=  $(SRC)/uts/common/gssapi/gssd.x
gssd.x: $(GSSDX)
        rm -f $@
        cp $(GSSDX) $@

#  Rules to generate derived rpcgen files from gssd.x spec file.

# NOTE WELL: There is code in gssd that assumes gssd is NOT
# multi-threaded.  Do NOT add -A to the rpcgen argument list in the
# Makefile unless you also remove this assumption.

gssd.h:         gssd.x
        $(RM) $@
        $(RPCGEN) -M -h gssd.x > $@

gssd_clnt.c:    gssd.x
        $(RM) $@
        $(RPCGEN) -M -l gssd.x > $@

gssd_svc.c:     gssd.x
        $(RM) $@
        $(RPCGEN) -M -m gssd.x > $@

gssd_xdr.c:     gssd.x
        $(RM) $@
```

```
128         $(RPCGEN) -M -c gssd.x > $@

130 $(OBJS): gssd.h

132 install: all $(DIRS) $(ROOTLIBPROG) $(ROOTMANIFEST)

134 install_h:

136 clean:
137         $(RM) $(OBJS) $(RSRC) gssd.x

139 lint_gssd:
140         $(LINT.c) $(GSSD_LINTS)

142 lint_gssc:
143         $(LINT.c) $(GSSC_LINTS)

145 lint:   lint_gssd lint_gssc

147 check:  $(CHKMANIFEST)

149 include ../../Makefile.targ

151 # EXPORT DELETE START
152 # Special targets to clean up the source tree for export distribution
153 # The WS target modifies the SCCS files as well, so a working workspace
154 # can be shipped.
155 # Warning: These targets change the source tree, the first only at the
156 #               plain source level, but the second changes the guts!
157 EXPORT_SRC:
158         $(RM) Makefile+ gssd_clnt_stubs.c+ gssd_proc.c+ gssdtest.c+
159         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
160                 < Makefile > Makefile+
161         $(MV) Makefile+ Makefile
162         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
163                 < gssd_clnt_stubs.c > gssd_clnt_stubs.c+
164         $(MV) gssd_clnt_stubs.c+ gssd_clnt_stubs.c
165         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
166                 < gssd_proc.c > gssd_proc.c+
167         $(MV) gssd_proc.c+ gssd_proc.c
168         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
169                 < gssdtest.c > gssdtest.c+
170         $(MV) gssdtest.c+ gssdtest.c
171         $(CHMOD) 444  Makefile gssd_clnt_stubs.c gssd_proc.c gssdtest.c

173 # EXPORT DELETE END

151 $(POFILE): $(DERIVED_FILES) .WAIT $(POFILES)
152         $(RM) $@
153         $(CAT) $(POFILES) > $@

155 generic.po: FRC
156         $(RM) messages.po
157         $(XGETTEXT) $(XGETFLAGS) `$(GREP) -l gettext *.[ch]`
158         $(SED) "/^domain/d" messages.po > $@
159         $(RM) messages.po

161 FRC:
```

```
**********************************************************
   67576 Thu Jul 11 01:28:53 2013
new/usr/src/cmd/gss/gssd/gssd_clnt_stubs.c
first pass
**********************************************************
______unchanged_portion_omitted_

1437 /* EXPORT DELETE START */

1437 OM_uint32
1438 kgss_seal_wrapped(
1439         minor_status,
1440         context_handle,
1441         conf_req_flag,
1442         qop_req,
1443         input_message_buffer,
1444         conf_state,
1445         output_message_buffer,
1446         gssd_context_verifier)

1448         OM_uint32 *minor_status;
1449         gssd_ctx_id_t context_handle;
1450         OM_uint32 gssd_context_verifier;
1451         int conf_req_flag;
1452         int qop_req;
1453         gss_buffer_t input_message_buffer;
1454         int *conf_state;
1455         gss_buffer_t output_message_buffer;
1456 {
1457         gss_seal_arg arg;
1458         gss_seal_res res;

1460         /* get the client handle to GSSD */

1462         if ((clnt = getgssd_handle()) == NULL) {
1463                 clnt_pcreateerror(server);
1464                 return (GSS_S_FAILURE);
1465         }

1467         /* copy the procedure arguments into the rpc arg parameter */


1470         arg.context_handle.GSS_CTX_ID_T_len = (uint_t)sizeof (gssd_ctx_id_t);
1471         arg.context_handle.GSS_CTX_ID_T_val = (char *)&context_handle;
1472         arg.gssd_context_verifier = gssd_context_verifier;

1474         arg.conf_req_flag = conf_req_flag;

1476         arg.qop_req = qop_req;

1478         arg.input_message_buffer.GSS_BUFFER_T_len =
1479                                 (uint_t)input_message_buffer->length;

1481         arg.input_message_buffer.GSS_BUFFER_T_val =
1482                                 (char *)input_message_buffer->value;

1484         /* call the remote procedure */

1486         memset(&res, 0, sizeof (res));
1487         if (gss_seal_1(&arg, &res, clnt) != RPC_SUCCESS) {

1489         /*
1490          * if the RPC call times out, null out all return arguments,
1491          * set minor_status to its maximum value, and return GSS_S_FAILURE
1492          */
```



```
1494                 if (minor_status != NULL)
1495                         *minor_status = DEFAULT_MINOR_STAT;
1496                 if (conf_state != NULL)
1497                         *conf_state = 0;
1498                 if (output_message_buffer != NULL)
1499                         output_message_buffer->length = 0;

1501                 return (GSS_S_FAILURE);
1502         }

1504         /* copy the rpc results into the return arguments */

1506         if (minor_status != NULL)
1507                 *minor_status = res.minor_status;

1509         if (conf_state != NULL)
1510                 *conf_state = res.conf_state;

1512         if (output_message_buffer != NULL) {
1513                 output_message_buffer->length =
1514                                 res.output_message_buffer.GSS_BUFFER_T_len;

1516                 output_message_buffer->value =
1517                                 (void *) MALLOC(output_message_buffer->length);
1518                 memcpy(output_message_buffer->value,
1519                         res.output_message_buffer.GSS_BUFFER_T_val,
1520                         output_message_buffer->length);
1521         }

1523         /*
1524          * free the memory allocated for the results and return with the status
1525          * received in the rpc call
1526          */

1528         clnt_freeres(clnt, xdr_gss_seal_res, (caddr_t)&res);
1529         return (res.status);
1530 }
______unchanged_portion_omitted_

1661 /* EXPORT DELETE END */

1659 OM_uint32
1660 kgss_display_status(minor_status,
1661                 status_value,
1662                 status_type,
1663                 mech_type,
1664                 message_context,
1665                 status_string,
1666                 uid)
1667         OM_uint32 *minor_status;
1668         OM_uint32 status_value;
1669         int status_type;
1670         gss_OID mech_type;
1671         int *message_context;
1672         gss_buffer_t status_string;
1673         uid_t uid;
1674 {
1675         gss_display_status_arg arg;
1676         gss_display_status_res res;

1678         /* get the client handle to GSSD */

1680         if ((clnt = getgssd_handle()) == NULL) {
1681                 clnt_pcreateerror(server);
1682                 return (GSS_S_FAILURE);
```

```
1683         }

1685         /* copy the procedure arguments into the rpc arg parameter */

1687         arg.uid = (OM_uint32) uid;

1689         arg.status_value = status_value;
1690         arg.status_type = status_type;

1692         arg.mech_type.GSS_OID_len = (uint_t)(mech_type != GSS_C_NULL_OID ?
1693                                         mech_type->length : 0);
1694         arg.mech_type.GSS_OID_val = (char *)(mech_type != GSS_C_NULL_OID ?
1695                                         mech_type->elements : 0);

1697         arg.message_context = *message_context;

1699         /* call the remote procedure */

1701         if (message_context != NULL)
1702                 *message_context = 0;
1703         if (status_string != NULL) {
1704                 status_string->length = 0;
1705                 status_string->value = NULL;
1706         }

1708         memset(&res, 0, sizeof (res));
1709         if (gss_display_status_1(&arg, &res, clnt) != RPC_SUCCESS) {

1711         /*
1712          * if the RPC call times out, null out all return arguments,
1713          * set minor_status to its maximum value, and return GSS_S_FAILURE
1714          */

1716                 if (minor_status != NULL)
1717                         *minor_status = DEFAULT_MINOR_STAT;

1719                 return (GSS_S_FAILURE);
1720         }

1722         if (minor_status != NULL)
1723                 *minor_status = res.minor_status;

1725         /* now process the results and pass them back to the caller */

1727         if (res.status == GSS_S_COMPLETE) {
1728                 if (message_context != NULL)
1729                         *message_context = res.message_context;
1730                 if (status_string != NULL) {
1731                         status_string->length =
1732                                 (size_t)res.status_string.GSS_BUFFER_T_len;
1733                         status_string->value =
1734                                 (void *)MALLOC(status_string->length);
1735                         memcpy(status_string->value,
1736                                 res.status_string.GSS_BUFFER_T_val,
1737                                 status_string->length);
1738                 }
1739         }

1741         clnt_freeres(clnt, xdr_gss_display_status_res, (caddr_t)&res);
1742         return (res.status);
1743 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   70235 Thu Jul 11 01:28:53 2013
new/usr/src/cmd/gss/gssd/gssd_proc.c
first pass
**********************************************************
______unchanged_portion_omitted_

1909 /* EXPORT DELETE START */

1909 bool_t
1910 gss_seal_1_svc(argp, res, rqstp)
1911 gss_seal_arg *argp;
1912 gss_seal_res *res;
1913 struct svc_req *rqstp;
1914 {
1915         uid_t uid;

1917         gss_buffer_desc input_message_buffer;
1918         gss_buffer_desc output_message_buffer;
1919         gss_ctx_id_t context_handle;
1920         bool_t context_verf_ok;

1922         memset(res, 0, sizeof (*res));

1924         if (gssd_debug)
1925                 fprintf(stderr, gettext("gss_seal\n"));

1927         gssd_convert_context_handle(&argp->context_handle, &context_handle,
1928                 argp->gssd_context_verifier, &context_verf_ok, NULL);

1930         /* verify the context_handle */

1932         if (!context_verf_ok) {
1933                 res->output_message_buffer.GSS_BUFFER_T_val = NULL;
1934                 res->output_message_buffer.GSS_BUFFER_T_len = 0;
1935                 res->status = (OM_uint32) GSS_S_NO_CONTEXT;
1936                 res->minor_status = 0;
1937                 return (TRUE);
1938         }

1940         /*
1941          * if the request isn't from root, null out the result pointer
1942          * entries, so the next time through xdr_free won't try to
1943          * free unmalloc'd memory and then return NULL
1944          */

1946         if (checkfrom(rqstp, &uid) == 0) {
1947                 res->output_message_buffer.GSS_BUFFER_T_val = NULL;
1948                 return (FALSE);

1950         }


1953         /*
1954          * copy the XDR structured arguments into their corresponding local
1955          * GSSAPI variable equivalents.
1956          */

1958         input_message_buffer.length = (size_t)argp->input_message_buffer.
1959                                       GSS_BUFFER_T_len;
1960         input_message_buffer.value = (void *)argp->input_message_buffer.
1961                                      GSS_BUFFER_T_val;


1964         /* call the gssapi routine */
```



```
1966         res->status = (OM_uint32)gss_seal(&res->minor_status,
1967                                           context_handle,
1968                                           argp->conf_req_flag,
1969                                           argp->qop_req,
1970                                           &input_message_buffer,
1971                                           &res->conf_state,
1972                                           &output_message_buffer);
1973         /*
1974          * convert the output args from the parameter given in the call to the
1975          * variable in the XDR result
1976          */

1978         if (res->status == GSS_S_COMPLETE) {
1979                 res->output_message_buffer.GSS_BUFFER_T_len =
1980                                           (uint_t)output_message_buffer.length;
1981                 res->output_message_buffer.GSS_BUFFER_T_val =
1982                                           (char *)output_message_buffer.value;
1983         } else
1984                 syslog_gss_error(res->status, res->minor_status, "seal");

1986 /* return to caller */

1988         return (TRUE);
1989 }
______unchanged_portion_omitted_

2075 /* EXPORT DELETE END */

2073 bool_t
2074 gss_display_status_1_svc(argp, res, rqstp)
2075 gss_display_status_arg *argp;
2076 gss_display_status_res *res;
2077 struct svc_req *rqstp;
2078 {
2079         uid_t uid;
2080         gss_OID mech_type;
2081         gss_OID_desc mech_type_desc;
2082         gss_buffer_desc status_string;

2084         memset(res, 0, sizeof (*res));

2086         if (gssd_debug)
2087                 fprintf(stderr, gettext("gss_display_status\n"));

2089         /*
2090          * if the request isn't from root, null out the result pointer
2091          * entries, so the next time through xdr_free won't try to
2092          * free unmalloc'd memory and then return NULL
2093          */

2095         if (checkfrom(rqstp, &uid) == 0) {
2096                 res->status_string.GSS_BUFFER_T_val = NULL;
2097                 return (FALSE);
2098         }

2100         /* set the uid sent as the RPC argument */

2102         uid = argp->uid;
2103         set_gssd_uid(uid);

2105         /*
2106          * copy the XDR structured arguments into their corresponding local
2107          * GSSAPI variables.
2108          */

2110         if (argp->mech_type.GSS_OID_len == 0)
```

```
2111                    mech_type = GSS_C_NULL_OID;
2112            else {
2113                    mech_type = &mech_type_desc;
2114                    mech_type_desc.length = (OM_uint32) argp->mech_type.GSS_OID_len;
2115                    mech_type_desc.elements = (void *) argp->mech_type.GSS_OID_val;
2116            }


2119            /* call the gssapi routine */

2121            res->status = (OM_uint32) gss_display_status(&res->minor_status,
2122                                            argp->status_value,
2123                                            argp->status_type,
2124                                            mech_type,
2125                                            (OM_uint32 *)&res->message_context,
2126                                            &status_string);

2128            /*
2129             * convert the output args from the parameter given in the call to the
2130             * variable in the XDR result
2131             */

2133            if (res->status == GSS_S_COMPLETE) {
2134                    res->status_string.GSS_BUFFER_T_len =
2135                            (uint_t)status_string.length;
2136                    res->status_string.GSS_BUFFER_T_val =
2137                            (char *)status_string.value;
2138            }

2140            return (TRUE);

2142 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   53226 Thu Jul 11 01:28:54 2013
new/usr/src/cmd/gss/gssd/gssdtest.c
first pass
**********************************************************
______unchanged_portion_omitted_

  83 #else /* !_KERNEL */
  84 #define OCTAL_MACRO "%03.3o."
  85 #define MALLOC(n) malloc(n)
  86 #define CALLOC(n, s) calloc((n), (s))
  87 #define FREE(x, n) free(x)
  88 #endif /* _KERNEL */

  90 static gss_OID gss_str2oid(char *);
  91 static char * gss_oid2str(gss_OID);
  92 static void instructs();
  93 static void usage();
  94 static int parse_input_line(char *, int *, char ***);
  95 extern uid_t getuid();

  97 static void _gss_init_sec_context(int, char **);
  98 static void _gss_acquire_cred(int, char **);
  99 static void _gss_add_cred(int, char **);
 100 static void _gss_sign(int, char **);
 101 static void _gss_release_cred(int, char **);
 102 static void _gss_accept_sec_context(int, char **);
 103 static void _gss_process_context_token(int, char **);
 104 static void _gss_delete_sec_context(int, char **);
 105 static void _gss_context_time(int, char **);
 106 static void _gss_verify(int, char **);
 107 /* EXPORT DELETE START */
 107 static void _gss_seal(int, char **);
 108 static void _gss_unseal(int, char **);
 110 /* EXPORT DELETE END */
 109 static void _gss_display_status(int, char **);
 110 static void _gss_indicate_mechs(int, char **);
 111 static void _gss_inquire_cred(int, char **);
 112 static void _gssd_expname_to_unix_cred(int, char **);
 113 static void _gssd_name_to_unix_cred(int, char **);
 114 static void _gssd_get_group_info(int, char **);

 116 static int do_gssdtest(char *buf);


 119 #ifndef _KERNEL
 120 static int read_line(char *buf, int size)
 121 {
 122         int len;

 124         /* read the next line. If cntl-d, return with zero char count */
 125         printf(gettext("\n> "));

 127         if (fgets(buf, size, stdin) == NULL)
 128                 return (0);

 130         len = strlen(buf);
 131         buf[--len] = '\0';
 132         return (len);
 133 }
______unchanged_portion_omitted_
 160 #endif /* !_KERNEL */

 162 static int
 163 do_gssdtest(char *buf)
 164 {
```



```
 165         int argc, seal_argc;
 166         int i;
 167         char **argv, **argv_array;

 169         char *cmd;
 170         char *seal_ini_array [] = { "initiator", " Hello"};
 171         char *seal_acc_array [] = { "acceptor", " Hello"};
 172         char *unseal_acc_array [] = {"acceptor"};
 173         char *unseal_ini_array [] = {"initiator"};
 174         char *delet_acc_array [] = {"acceptor"};
 175         char *delet_ini_array [] = {"initiator"};

 177         argv = 0;

 179         if (parse_input_line(buf, &argc, &argv) == 0) {
 180                 printf(gettext("\n"));
 181                 return (1);
 182         }

 184         if (argc == 0) {
 185                 usage();
 186                 /*LINTED*/
 187                 FREE(argv_array, (argc+1)*sizeof (char *));
 188                 return (0);
 189         }

 191         /*
 192          * remember argv_array address, which is memory calloc'd by
 193          * parse_input_line, so it can be free'd at the end of the loop.
 194          */

 196         argv_array = argv;

 198         cmd = argv[0];

 200         argc--;
 201         argv++;

 203         if (strcmp(cmd, "gss_loop") == 0 ||
 204             strcmp(cmd, "loop") == 0) {

 206                 if (argc < 1) {
 207                         usage();
 208                         FREE(argv_array, (argc+2) * sizeof (char *));
 209                         return (0);
 210                 }
 211                 for (i = 0; i < LOOP_COUNTER; i++) {
 212                         printf(gettext("Loop Count is %d \n"), i);
 213                         /*
 214                          * if (i > 53)
 215                          *      printf ("Loop counter is greater than 55\n");
 216                          */
 217                         _gss_acquire_cred(argc, argv);
 218                         _gss_init_sec_context(argc, argv);
 219                         _gss_accept_sec_context(0, argv);
 220                         _gss_init_sec_context(argc, argv);

 223 /* EXPORT DELETE START */
 222                         seal_argc = 2;
 223                         _gss_seal(seal_argc, seal_ini_array);
 224                         seal_argc = 1;
 225                         _gss_unseal(seal_argc, unseal_acc_array);
 226                         seal_argc = 2;
 227                         _gss_seal(seal_argc, seal_acc_array);
 228                         seal_argc = 1;
 229                         _gss_unseal(seal_argc, unseal_ini_array);
```

```
/* EXPORT DELETE END */
                        seal_argc = 2;
                        _gss_sign(seal_argc, seal_ini_array);
                        seal_argc = 1;
                        _gss_verify(seal_argc, unseal_acc_array);
                        seal_argc = 2;
                        _gss_sign(seal_argc, seal_acc_array);
                        seal_argc = 1;
                        _gss_verify(seal_argc, unseal_ini_array);
                        _gss_delete_sec_context(argc, delet_acc_array);
                        _gss_delete_sec_context(argc, delet_ini_array);
                }
        }
        if (strcmp(cmd, "gss_all") == 0 ||
            strcmp(cmd, "all") == 0) {
                _gss_acquire_cred(argc, argv);
                _gss_init_sec_context(argc, argv);
                _gss_accept_sec_context(0, argv);
                _gss_init_sec_context(argc, argv);

/* EXPORT DELETE START */
                seal_argc = 2;
                _gss_seal(seal_argc, seal_acc_array);
                seal_argc = 1;
                _gss_unseal(seal_argc, unseal_ini_array);
                seal_argc = 2;
                _gss_seal(seal_argc, seal_ini_array);
                seal_argc = 1;
                _gss_unseal(seal_argc, unseal_acc_array);
/* EXPORT DELETE END */
                seal_argc = 2;
                _gss_sign(seal_argc, seal_ini_array);
                seal_argc = 1;
                _gss_verify(seal_argc, unseal_acc_array);
                seal_argc = 2;
                _gss_sign(seal_argc, seal_acc_array);
                seal_argc = 1;
                _gss_verify(seal_argc, unseal_ini_array);

        }
        if (strcmp(cmd, "gss_acquire_cred") == 0 ||
            strcmp(cmd, "acquire") == 0) {
                _gss_acquire_cred(argc, argv);
                if (argc == 1)
                        _gss_add_cred(argc, argv);
        }

        else if (strcmp(cmd, "gss_release_cred") == 0 ||
                strcmp(cmd, "release") == 0)
                _gss_release_cred(argc, argv);
        else if (strcmp(cmd, "gss_init_sec_context") == 0 ||
                strcmp(cmd, "init") == 0)
                _gss_init_sec_context(argc, argv);
        else if (strcmp(cmd, "gss_accept_sec_context") == 0 ||
                strcmp(cmd, "accept") == 0)
                _gss_accept_sec_context(argc, argv);
        else if (strcmp(cmd, "gss_process_context_token") == 0 ||
                strcmp(cmd, "process") == 0)
                _gss_process_context_token(argc, argv);
        else if (strcmp(cmd, "gss_delete_sec_context") == 0 ||
                strcmp(cmd, "delete") == 0)
                _gss_delete_sec_context(argc, argv);
        else if (strcmp(cmd, "gss_context_time") == 0 ||
                strcmp(cmd, "time") == 0)
                _gss_context_time(argc, argv);
        else if (strcmp(cmd, "gss_sign") == 0 ||
```



```
                strcmp(cmd, "sign") == 0)
                _gss_sign(argc, argv);
        else if (strcmp(cmd, "gss_verify") == 0 ||
                strcmp(cmd, "verify") == 0)
                _gss_verify(argc, argv);
/* EXPORT DELETE START */
        else if (strcmp(cmd, "gss_seal") == 0 ||
                strcmp(cmd, "seal") == 0)
                _gss_seal(argc, argv);
        else if (strcmp(cmd, "gss_unseal") == 0 ||
                strcmp(cmd, "unseal") == 0)
                _gss_unseal(argc, argv);
/* EXPORT DELETE END */
        else if (strcmp(cmd, "gss_display_status") == 0||
                strcmp(cmd, "status") == 0)
                _gss_display_status(argc, argv);
        else if (strcmp(cmd, "gss_indicate_mechs") == 0 ||
                strcmp(cmd, "indicate") == 0)
                _gss_indicate_mechs(argc, argv);
        else if (strcmp(cmd, "gss_inquire_cred") == 0 ||
                strcmp(cmd, "inquire") == 0)
                _gss_inquire_cred(argc, argv);
        else if (strcmp(cmd, "expname2unixcred") == 0 ||
                strcmp(cmd, "gsscred_expname_to_unix_cred") == 0)
                _gssd_expname_to_unix_cred(argc, argv);
        else if (strcmp(cmd, "name2unixcred") == 0 ||
                strcmp(cmd, "gsscred_name_to_unix_cred") == 0)
                _gssd_name_to_unix_cred(argc, argv);
        else if (strcmp(cmd, "grpinfo") == 0 ||
                strcmp(cmd, "gss_get_group_info") == 0)
                _gssd_get_group_info(argc, argv);
        else if (strcmp(cmd, "exit") == 0) {
                printf(gettext("\n"));
                FREE(argv_array, (argc+2) * sizeof (char *));
                return (1);
        } else
                usage();

        /* free argv array */

        FREE(argv_array, (argc+2) * sizeof (char *));
        return (0);
}
______unchanged_portion_omitted_

/* EXPORT DELETE START */
static void
_gss_seal(argc, argv)
int argc;
char **argv;
{
        OM_UINT32 status;

        OM_uint32 minor_status;
        gss_ctx_id_t context_handle;
        int conf_req_flag;
        int qop_req;
        gss_buffer_desc input_message_buffer;
        int conf_state;
        uid_t uid;

        uid = (uid_t) getuid();

        /*
         * specify the default confidentiality requested (both integrity
         * and confidentiality) and quality of protection
```

```
1537          */

1539         conf_req_flag = 1;
1540         qop_req = GSS_C_QOP_DEFAULT;

1542         /* set up the arguments specified in the input parameters */

1544         if (argc == 0) {
1545                 usage();
1546                 return;
1547         }


1550         if (strcmp(argv[0], "initiator") == 0)
1551                 context_handle = initiator_context_handle;
1552         else if (strcmp(argv[0], "acceptor") == 0)
1553                 context_handle = acceptor_context_handle;
1554         else {
1555                 printf(gettext(
1556                         "must specify either \"initiator\" or \"acceptor\"\n"));
1557                 return;
1558         }

1560         argc--;
1561         argv++;

1563         if (argc == 0) {
1564                 usage();
1565                 return;
1566         }


1569         input_message_buffer.length = strlen(argv[0])+1;
1570         input_message_buffer.value =
1571                 (void *) MALLOC(input_message_buffer.length);
1572         strcpy(input_message_buffer.value, argv[0]);

1574         argc--;
1575         argv++;

1577         if (argc != 0) {
1578                 usage();
1579                 return;
1580         }

1582         status = kgss_seal(&minor_status,
1583                         context_handle,
1584                         conf_req_flag,
1585                         qop_req,
1586                         &input_message_buffer,
1587                         &conf_state,
1588                         &message_buffer,
1589                         uid);

1591         /* store major and minor status for gss_display_status() call */

1593         gss_major_code = status;
1594         gss_minor_code = minor_status;

1596         /* free the inputmessage buffer */

1598         gss_release_buffer(&minor_status, &input_message_buffer);

1600         if (status != GSS_S_COMPLETE) {
1601                 printf(gettext("server ret err (octal) %o (%s)\n"),
1602                         status, gettext("gss_seal error"));
```



```
1603                 return;
1604         } else {
1605                 printf(gettext("\nseal succeeded\n\n"));
1606                 return;
1607         }
1608 }
______unchanged_portion_omitted_
1691 /* EXPORT DELETE END */

1685 static void
1686 _gss_display_status(argc, argv)
1687 int argc;
1688 char **argv;
1689 {
1690         OM_UINT32 status;
1691         OM_uint32 minor_status;
1692         int status_type;
1693         int status_value;
1694         gss_OID mech_type = (gss_OID) 0;
1695         int message_context;
1696         gss_buffer_desc status_string;
1697         uid_t uid;

1699         uid = (uid_t) getuid();

1701         /* initialize message context to zero */

1703         message_context = 0;

1705         if (argc == 0) {
1706                 printf(gettext("Assuming Kerberos V5 as the mechanism\n"));
1707                 printf(gettext(
1708                         "The mech OID 1.2.840.113554.1.2.2 will be used\n"));
1709                 mech_type = gss_str2oid((char *)GSS_KRB5_MECH_OID);
1710         } else
1711                 mech_type = gss_str2oid(argv[0]);

1713         if (mech_type == 0 || mech_type->length == 0) {
1714                 printf(gettext("improperly formated mechanism OID\n"));
1715                 return;
1716         }

1718         /* Is this call for the major or minor status? */

1720         if (strcmp(argv[0], "major") == 0) {
1721                 status_type = GSS_C_GSS_CODE;
1722                 status_value = gss_major_code;
1723         } else if (strcmp(argv[0], "minor") == 0) {
1724                 status_type = GSS_C_MECH_CODE;
1725                 status_value = gss_minor_code;
1726         } else {
1727                 printf(gettext("must specify either \"major\" or \"minor\"\n"));
1728                 return;
1729         }

1731         argc--;
1732         argv++;

1734         if (argc != 0) {
1735                 usage();
1736                 return;
1737         }

1739         status = kgss_display_status(&minor_status,
1740                                 status_value,
1741                                 status_type,
```

```
1742                                mech_type,
1743                                &message_context,
1744                                &status_string,
1745                                uid);

1747        if (status == GSS_S_COMPLETE) {
1748                printf(gettext("status =\n  %s\n\n"), status_string.value);
1749        } else if (status == GSS_S_BAD_MECH) {
1750                printf(gettext("invalide mechanism OID\n\n"));
1751        } else {
1752                printf(gettext("server ret err (octal) %o (%s)\n"),
1753                        status, gettext("gss_display_status error"));
1754        }
1755 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1476 Thu Jul 11 01:28:55 2013
new/usr/src/cmd/krb5/kadmin/Makefile
first pass
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 # Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
  22 # Use is subject to license terms.
  23 #
  24 # cmd/krb5/kadmin/Makefile

  26 include ../../Makefile.cmd

  28 SUBDIRS= cli dbutil ktutil kpasswd server kclient kdcmgr gui
  28 SUBDIRS= cli dbutil ktutil kpasswd server kclient kdcmgr
  29 # EXPORT DELETE START
  30 SUBDIRS += gui
  31 # EXPORT DELETE END

  30 all :=          TARGET= all
  31 clean :=        TARGET= clean
  32 clobber :=      TARGET= clobber
  33 delete :=       TARGET= delete
  34 install :=      TARGET= install
  35 lint :=         TARGET= lint
  36 catalog :=      TARGET= catalog
  37 package :=      TARGET= package
  38 _msg:=          TARGET= _msg

  40 _msg:   $(SUBDIRS)

  42 .KEEP_STATE:

  44 all clean clobber delete install lint catalog package: $(SUBDIRS)

  46 # install rule for install_h target

  48 install: $(SUBDIRS)

  50 check: $(CHECKHDRS)

  52 $(SUBDIRS):     FRC
  53         @cd $@; pwd; $(MAKE) $(TARGET)

  55 FRC:

  60 # EXPORT DELETE START
```



```
  61 EXPORT_SRC:
  62         $(RM) -r gui
  63         $(RM) Makefile+
  64         $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
  65                 < Makefile > Makefile+
  66         $(MV) Makefile+ Makefile
  67         $(CHMOD) 444 Makefile
  68 # EXPORT DELETE END
```

```
**********************************************************
    1752 Thu Jul 11 01:28:56 2013
new/usr/src/cmd/login/Makefile
onc plus-be-gone
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #

  26 PROG= login
  27 OBJS= login.o login_audit.o
  28 SRCS= $(OBJS:%.o=%.c)
  29 ONC_SRCS=$(SRCS:%.c=%.c_onc_plus)

  30 DEFAULTFILES= login.dfl

  32 include ../Makefile.cmd

  34 LOGINDEVPERM= logindevperm
  35 LOGINDEVPERMSRC= $(LOGINDEVPERM).sh
  36 ROOTLOGINDEVPERM= $(LOGINDEVPERM:%=$(ROOTETC)/%)

  38 $(ROOTLOGINDEVPERM) := FILEMODE = 644
  39 FILEMODE= 4555

  41 CLOBBERFILES += $(LOGINDEVPERM)
  42 CLOBBERFILES += $(LOGINDEVPERM) $(ONC_SRCS)

  43 CPPFLAGS +=     -DSYSV -DCONSOLE='"/dev/console"' -DSECURITY \
  44                 -D_FILE_OFFSET_BITS=64 -I$(SRC)/lib/pam_modules/krb5

  46 LDLIBS += -lbsm -lpam -ldevinfo
  47 CFLAGS += $(CCVERBOSE)

  49 .KEEP_STATE:


  52 all: $(PROG) $(ROOTLOGINDEVPERM)

  54 $(LOGINDEVPERM): $(LOGINDEVPERMSRC)
  55         $(RM) $(LOGINDEVPERM)
  56         /bin/sh $(LOGINDEVPERMSRC) > $(LOGINDEVPERM)

  58 $(PROG): $(OBJS)
  59         $(LINK.c) $(OBJS) -o $@ $(LDLIBS)
```



```
  60         $(POST_PROCESS)

  62 install: all $(DIRS) $(ROOTPROG) $(ROOTETCDEFAULTFILES) $(ROOTLOGINDEVPERM)

  64 clean:
  65         $(RM) $(OBJS)

  67 lint:   lint_SRCS

  69 include ../Makefile.targ

  72 # make ONC_PLUS using suffix rule
  73 #

  75 ONC_PLUS: $(ONC_SRCS)
```

```
**********************************************************
   56556 Thu Jul 11 01:28:56 2013
new/usr/src/cmd/login/login.c
onc plus-be-gone
**********************************************************
```

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

/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/* ONC_PLUS EXTRACT START */
/*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */

/*
 * University Copyright- Copyright (c) 1982, 1986, 1988
 * The Regents of the University of California
 * All Rights Reserved
 *
 * University Acknowledgment- Portions of this document are derived from
 * software developed by the University of California, Berkeley, and its
 * contributors.
 */

/*      Copyright (c) 1987, 1988 Microsoft Corporation  */
/*        All Rights Reserved   */

/* ONC_PLUS EXTRACT END */

/*
 * For a complete reference to login(1), see the manual page.  However,
 * login has accreted some intentionally undocumented options, which are
 * explained here:
 *
 * -a: This legacy flag appears to be unused.
 *
 * -f <username>: This flag was introduced by PSARC 1995/039 in support
 *    of Kerberos.  But it's not used by Sun's Kerberos implementation.
 *    It is however employed by zlogin(1), since it allows one to tell
 *    login: "This user is authenticated."  In the case of zlogin that's
 *    true because the zone always trusts the global zone.
 *
 * -z <zonename>: This flag is passed to login when zlogin(1) executes a
 *    zone login.  This tells login(1) to skip it's normal CONSOLE check
 *    (i.e. that the root login must be on /dev/console) and tells us the
```



```
 *    name of the zone from which the login is occurring.
 */

#include <sys/types.h>
#include <sys/param.h>
#include <unistd.h>     /* For logfile locking */
#include <signal.h>
#include <stdio.h>
#include <sys/stat.h>
#include <string.h>
#include <deflt.h>
#include <grp.h>
#include <fcntl.h>
#include <lastlog.h>
#include <termio.h>
#include <utmpx.h>
#include <stdlib.h>
#include <wait.h>
#include <errno.h>
#include <ctype.h>
#include <syslog.h>
#include <ulimit.h>
#include <libgen.h>
#include <pwd.h>
#include <security/pam_appl.h>
#include <strings.h>
#include <libdevinfo.h>
#include <zone.h>
#include "login_audit.h"

#include <krb5_repository.h>
/*
 *
 *          *** Defines, Macros, and String Constants  ***
 *
 *
 */

#define ISSUEFILE "/etc/issue"  /* file to print before prompt */
#define NOLOGIN "/etc/nologin"  /* file to lock users out during shutdown */

/*
 * These need to be defined for UTMPX management.
 * If we add in the utility functions later, we
 * can remove them.
 */
#define __UPDATE_ENTRY  1
#define __LOGIN         2

/*
 * Intervals to sleep after failed login
 */
#ifndef SLEEPTIME
#define SLEEPTIME 4     /* sleeptime before login incorrect msg */
#endif
static int      Sleeptime = SLEEPTIME;

/*
 * seconds login disabled after allowable number of unsuccessful attempts
 */
#ifndef DISABLETIME
#define DISABLETIME     20
#endif
static int      Disabletime = DISABLETIME;

#define MAXTRYS         5
```

```
126 static int      retry = MAXTRYS;

128 /*
129  * Login logging support
130  */
131 #define LOGINLOG        "/var/adm/loginlog"     /* login log file */
132 #define LNAME_SIZE      20      /* size of logged logname */
133 #define TTYN_SIZE       15      /* size of logged tty name */
134 #define TIME_SIZE       30      /* size of logged time string */
135 #define ENT_SIZE        (LNAME_SIZE + TTYN_SIZE + TIME_SIZE + 3)
136 #define L_WAITTIME      5       /* waittime for log file to unlock */
137 #define LOGTRYS         10      /* depth of 'try' logging */

139 /*
140  * String manipulation macros: SCPYN, SCPYL, EQN and ENVSTRNCAT
141  * SCPYL is the safer version of SCPYN
142  */
143 #define SCPYL(a, b)     (void) strlcpy(a, b, sizeof (a))
144 #define SCPYN(a, b)     (void) strncpy(a, b, sizeof (a))
145 #define EQN(a, b)       (strncmp(a, b, sizeof (a)-1) == 0)
146 #define ENVSTRNCAT(to, from) {int deflen; deflen = strlen(to); \
147         (void) strncpy((to)+ deflen, (from), sizeof (to) - (1 + deflen)); }

149 /*
150  * Other macros
151  */
152 #define NMAX    sizeof (((struct utmpx *)0)->ut_name)
153 #define HMAX    sizeof (((struct utmpx *)0)->ut_host)
154 #define min(a, b)       (((a) < (b)) ? (a) : (b))

156 /*
157  * Various useful files and string constants
158  */
159 #define SHELL           "/usr/bin/sh"
160 #define SHELL2          "/sbin/sh"
161 #define SUBLOGIN        "<!sublogin>"
162 #define LASTLOG         "/var/adm/lastlog"
163 #define PROG_NAME       "login"
164 #define HUSHLOGIN       ".hushlogin"

169 /* ONC_PLUS EXTRACT START */
166 /*
167  * Array and Buffer sizes
168  */
169 #define PBUFSIZE 8      /* max significant characters in a password */
174 /* ONC_PLUS EXTRACT END */
170 #define MAXARGS 63      /* change value below if changing this */
171 #define MAXARGSWIDTH 2  /* log10(MAXARGS) */
172 #define MAXENV 1024
173 #define MAXLINE 2048

175 /*
176  * Miscellaneous constants
177  */
178 #define ROOTUID         0
179 #define ERROR           1
180 #define OK              0
181 #define LOG_ERROR       1
182 #define DONT_LOG_ERROR  0
183 #define TRUE            1
184 #define FALSE           0

186 /*
187  * Counters for counting the number of failed login attempts
188  */
```



```
189 static int trys = 0;
190 static int count = 1;

192 /*
193  * error value for login_exit() audit output (0 == no audit record)
194  */
195 static int      audit_error = 0;

197 /*
198  * Externs a plenty
199  */
205 /* ONC_PLUS EXTRACT START */
200 extern  int     getsecretkey();
207 /* ONC_PLUS EXTRACT START */

202 /*
203  * The current user name
204  */
205 static  char    user_name[NMAX];
206 static  char    minusnam[16] = "-";

208 /*
209  * login_pid, used to find utmpx entry to update.
210  */
211 static pid_t    login_pid;

213 /*
214  * locale environments to be passed to shells.
215  */
216 static char *localeenv[] = {
217         "LANG",
218         "LC_CTYPE", "LC_NUMERIC", "LC_TIME", "LC_COLLATE",
219         "LC_MONETARY", "LC_MESSAGES", "LC_ALL", 0};
220 static int locale_envmatch(char *, char *);

222 /*
223  * Environment variable support
224  */
225 static  char    shell[256] = { "SHELL=" };
226 static  char    home[MAXPATHLEN] = { "HOME=" };
227 static  char    term[64] = { "TERM=" };
228 static  char    logname[30] = { "LOGNAME=" };
229 static  char    timez[100] = { "TZ=" };
230 static  char    hertz[10] = { "HZ=" };
231 static  char    path[MAXPATHLEN] = { "PATH=" };
232 static  char    *newenv[10+MAXARGS] =
233         {home, path, logname, hertz, term, 0, 0};
234 static  char    **envinit = newenv;
235 static  int     basicenv;
236 static  char    *zero = (char *)0;
237 static  char    **envp;
238 #ifndef NO_MAIL
239 static  char    mail[30] = { "MAIL=/var/mail/" };
240 #endif
241 extern char **environ;
242 static  char inputline[MAXLINE];

244 #define MAX_ID_LEN 256
245 #define MAX_REPOSITORY_LEN 256
246 #define MAX_PAMSERVICE_LEN 256

248 static char identity[MAX_ID_LEN];
249 static char repository[MAX_REPOSITORY_LEN];
250 static char progname[MAX_PAMSERVICE_LEN];
```

```
253 /*
254  * Strings used to prompt the user.
255  */
256 static  char    loginmsg[] = "login: ";
257 static  char    passwdmsg[] = "Password:";
258 static  char    incorrectmsg[] = "Login incorrect\n";

267 /* ONC_PLUS EXTRACT START */
260 /*
261  * Password file support
262  */
263 static  struct  passwd *pwd = NULL;
264 static  char    remote_host[HMAX];
265 static  char    zone_name[ZONENAME_MAX];

267 /*
268  * Illegal passwd entries.
269  */
270 static  struct  passwd nouser = { "", "no:password", (uid_t)-1 };
279 /* ONC_PLUS EXTRACT END */

272 /*
273  * Log file support
274  */
275 static  char    *log_entry[LOGTRYS];
276 static  int     writelog = 0;
277 static  int     lastlogok = 0;
278 static  struct lastlog ll;
279 static  int     dosyslog = 0;
280 static  int     flogin = MAXTRYS;       /* flag for SYSLOG_FAILED_LOGINS */

282 /*
283  * Default file toggles
284  */
285 static  char    *Pndefault      = "/etc/default/login";
286 static  char    *Altshell       = NULL;
287 static  char    *Console        = NULL;
288 static  int     Passreqflag     = 0;

290 #define DEFUMASK        022
291 static  mode_t  Umask           = DEFUMASK;
292 static  char    *Def_tz         = NULL;
293 static  char    *tmp_tz         = NULL;
294 static  char    *Def_hertz      = NULL;
295 #define SET_FSIZ        2                       /* ulimit() command arg */
296 static  long    Def_ulimit      = 0;
297 #define MAX_TIMEOUT     (15 * 60)
298 #define DEF_TIMEOUT     (5 * 60)
299 static  unsigned Def_timeout    = DEF_TIMEOUT;
300 static  char    *Def_path       = NULL;
301 static  char    *Def_supath     = NULL;
302 #define DEF_PATH        "/usr/bin:"     /* same as PATH */
303 #define DEF_SUPATH      "/usr/sbin:/usr/bin" /* same as ROOTPATH */

305 /*
306  * Defaults for updating expired passwords
307  */
308 #define DEF_ATTEMPTS    3

310 /*
311  * ttyprompt will point to the environment variable TTYPROMPT.
312  * TTYPROMPT is set by ttymon if ttymon already wrote out the prompt.
313  */
314 static  char    *ttyprompt = NULL;
315 static  char    *ttyn = NULL;
```



```
317 /*
318  * Pass inherited environment.  Used by telnetd in support of the telnet
319  * ENVIRON option.
320  */
321 static  boolean_t pflag = B_FALSE;
322 static  boolean_t uflag = B_FALSE;
323 static  boolean_t Rflag = B_FALSE;
324 static  boolean_t sflag = B_FALSE;
325 static  boolean_t Uflag = B_FALSE;
326 static  boolean_t tflag = B_FALSE;
327 static  boolean_t hflag = B_FALSE;
328 static  boolean_t rflag = B_FALSE;
329 static  boolean_t zflag = B_FALSE;

331 /*
332  * Remote login support
333  */
334 static  char    rusername[NMAX+1], lusername[NMAX+1];
335 static  char    terminal[MAXPATHLEN];

346 /* ONC_PLUS EXTRACT START */
337 /*
338  * Pre-authentication flag support
339  */
340 static  int     fflag;

342 static char ** getargs(char *);

344 static int login_conv(int, struct pam_message **,
345     struct pam_response **, void *);

347 static struct pam_conv pam_conv = {login_conv, NULL};
348 static pam_handle_t *pamh;      /* Authentication handle */
359 /* ONC_PLUS EXTRACT END */

350 /*
351  * Function declarations
352  */
353 static  void    turn_on_logging(void);
354 static  void    defaults(void);
355 static  void    usage(void);
356 static  void    process_rlogin(void);
368 /* ONC_PLUS EXTRACT START */
357 static  void    login_authenticate();
358 static  void    setup_credentials(void);
371 /* ONC_PLUS EXTRACT END */
359 static  void    adjust_nice(void);
360 static  void    update_utmpx_entry(int);
361 static  void    establish_user_environment(char **);
362 static  void    print_banner(void);
363 static  void    display_last_login_time(void);
364 static  void    exec_the_shell(void);
365 static  int     process_chroot_logins(void);
366 static  void    chdir_to_dir_user(void);
367 static  void    check_log(void);
368 static  void    validate_account(void);
369 static  void    doremoteterm(char *);
370 static  int     get_options(int, char **);
371 static  void    getstr(char *, int, char *);
372 static  int     legalenvvar(char *);
373 static  void    check_for_console(void);
374 static  void    check_for_dueling_unix(char *);
375 static  void    get_user_name(void);
376 static  uint_t  get_audit_id(void);
377 static  void    login_exit(int)__NORETURN;
378 static  int     logins_disabled(char *);
```

```
379 static  void    log_bad_attempts(void);
380 static  int     is_number(char *);

395 /* ONC_PLUS EXTRACT START */
382 /*
383  *                      *** main ***
384  *
385  *      The primary flow of control is directed in this routine.
386  *      Control moves in line from top to bottom calling subfunctions
387  *      which perform the bulk of the work.  Many of these calls exit
388  *      when a fatal error is encountered and do not return to main.
389  *
390  *
391  */

393 int
394 main(int argc, char *argv[], char **renvp)
395 {
410 /* ONC_PLUS EXTRACT END */
396         int sublogin;
397         int pam_rc;

399         login_pid = getpid();

401         /*
402          * Set up Defaults and flags
403          */
404         defaults();
405         SCPYL(progname, PROG_NAME);

407         /*
408          * Set up default umask
409          */
410         if (Umask > ((mode_t)0777))
411                 Umask = DEFUMASK;
412         (void) umask(Umask);

414         /*
415          * Set up default timeouts and delays
416          */
417         if (Def_timeout > MAX_TIMEOUT)
418                 Def_timeout = MAX_TIMEOUT;
419         if (Sleeptime < 0 || Sleeptime > 5)
420                 Sleeptime = SLEEPTIME;

422         (void) alarm(Def_timeout);

424         /*
425          * Ignore SIGQUIT and SIGINT and set nice to 0
426          */
427         (void) signal(SIGQUIT, SIG_IGN);
428         (void) signal(SIGINT, SIG_IGN);
429         (void) nice(0);

431         /*
432          * Set flag to disable the pid check if you find that you are
433          * a subsystem login.
434          */
435         sublogin = 0;
436         if (*renvp && strcmp(*renvp, SUBLOGIN) == 0)
437                 sublogin = 1;

439         /*
440          * Parse Arguments
441          */
442         if (get_options(argc, argv) == -1) {
```



```
443                 usage();
444                 audit_error = ADT_FAIL_VALUE_BAD_CMD;
445                 login_exit(1);
446         }

448         /*
449          * if devicename is not passed as argument, call ttyname(0)
450          */
451         if (ttyn == NULL) {
452                 ttyn = ttyname(0);
453                 if (ttyn == NULL)
454                         ttyn = "/dev/???";
455         }

472 /* ONC_PLUS EXTRACT START */
457         /*
458          * Call pam_start to initiate a PAM authentication operation
459          */

461         if ((pam_rc = pam_start(progname, user_name, &pam_conv, &pamh))
462             != PAM_SUCCESS) {
463                 audit_error = ADT_FAIL_PAM + pam_rc;
464                 login_exit(1);
465         }
466         if ((pam_rc = pam_set_item(pamh, PAM_TTY, ttyn)) != PAM_SUCCESS) {
467                 audit_error = ADT_FAIL_PAM + pam_rc;
468                 login_exit(1);
469         }
470         if ((pam_rc = pam_set_item(pamh, PAM_RHOST, remote_host)) !=
471             PAM_SUCCESS) {
472                 audit_error = ADT_FAIL_PAM + pam_rc;
473                 login_exit(1);
474         }

476         /*
477          * We currently only support special handling of the KRB5 PAM repository
478          */
479         if ((Rflag && strlen(repository)) &&
480             strcmp(repository, KRB5_REPOSITORY_NAME) == 0 &&
481             (uflag && strlen(identity))) {
482                 krb5_repository_data_t krb5_data;
483                 pam_repository_t pam_rep_data;

485                 krb5_data.principal = identity;
486                 krb5_data.flags = SUNW_PAM_KRB5_ALREADY_AUTHENTICATED;

488                 pam_rep_data.type = repository;
489                 pam_rep_data.scope = (void *)&krb5_data;
490                 pam_rep_data.scope_len = sizeof (krb5_data);

492                 (void) pam_set_item(pamh, PAM_REPOSITORY,
493                     (void *)&pam_rep_data);
494         }
511 /* ONC_PLUS EXTRACT END */

496         /*
497          * Open the log file which contains a record of successful and failed
498          * login attempts
499          */
500         turn_on_logging();

502         /*
503          * say "hi" to syslogd ..
504          */
505         openlog("login", 0, LOG_AUTH);
```

```
507         /*
508          * Do special processing for -r (rlogin) flag
509          */
510         if (rflag)
511                 process_rlogin();

530 /* ONC_PLUS EXTRACT START */
513         /*
514          * validate user
515          */
516         /* we are already authenticated. fill in what we must, then continue */
517         if (fflag) {
536 /* ONC_PLUS EXTRACT END */
518                 if ((pwd = getpwnam(user_name)) == NULL) {
519                         audit_error = ADT_FAIL_VALUE_USERNAME;

521                         log_bad_attempts();
522                         (void) printf("Login failed: unknown user '%s'.\n",
523                             user_name);
524                         login_exit(1);
525                 }
545 /* ONC_PLUS EXTRACT START */
526         } else {
527                 /*
528                  * Perform the primary login authentication activity.
529                  */
530                 login_authenticate();
531         }
552 /* ONC_PLUS EXTRACT END */

533         /* change root login, then we exec another login and try again */
534         if (process_chroot_logins() != OK)
535                 login_exit(1);

537         /*
538          * If root login and not on system console then call exit(2)
539          */
540         check_for_console();

542         /*
543          * Check to see if a shutdown is in progress, if it is and
544          * we are not root then throw the user off the system
545          */
546         if (logins_disabled(user_name) == TRUE) {
547                 audit_error = ADT_FAIL_VALUE_LOGIN_DISABLED;
548                 login_exit(1);
549         }

551         if (pwd->pw_uid == 0) {
552                 if (Def_supath != NULL)
553                         Def_path = Def_supath;
554                 else
555                         Def_path = DEF_SUPATH;
556         }

558         /*
559          * Check account expiration and passwd aging
560          */
561         validate_account();

563         /*
564          * We only get here if we've been authenticated.
565          */

567         /*
568          * Now we set up the environment for the new user, which includes
```



```
569          * the users ulimit, nice value, ownership of this tty, uid, gid,
570          * and environment variables.
571          */
572         if (Def_ulimit > 0L && ulimit(SET_FSIZ, Def_ulimit) < 0L)
573                 (void) printf("Could not set ULIMIT to %ld\n", Def_ulimit);

575         /* di_devperm_login() sends detailed errors to syslog */
576         if (di_devperm_login((const char *)ttyn, pwd->pw_uid, pwd->pw_gid,
577             NULL) == -1) {
578                 (void) fprintf(stderr, "error processing /etc/logindevperm,"
579                     " see syslog for more details\n");
580         }

582         adjust_nice();          /* passwd file can specify nice value */

605 /* ONC_PLUS EXTRACT START */
584         setup_credentials();    /* Set user credentials  - exits on failure */

586         /*
587          * NOTE: telnetd and rlogind rely upon this updating of utmpx
588          * to indicate that the authentication completed  successfully,
589          * pam_open_session was called and therefore they are required to
590          * call pam_close_session.
591          */
592         update_utmpx_entry(sublogin);

594         /* set the real (and effective) UID */
595         if (setuid(pwd->pw_uid) == -1) {
596                 login_exit(1);
597         }

599         /*
600          * Set up the basic environment for the exec.  This includes
601          * HOME, PATH, LOGNAME, SHELL, TERM, TZ, HZ, and MAIL.
602          */
603         chdir_to_dir_user();

605         establish_user_environment(renvp);

607         (void) pam_end(pamh, PAM_SUCCESS);      /* Done using PAM */
608         pamh = NULL;
631 /* ONC_PLUS EXTRACT END */

610         if (pwd->pw_uid == 0) {
611                 if (dosyslog) {
612                         if (remote_host[0]) {
613                                 syslog(LOG_NOTICE, "ROOT LOGIN %s FROM %.*s",
614                                     ttyn, HMAX, remote_host);
615                         } else
616                                 syslog(LOG_NOTICE, "ROOT LOGIN %s", ttyn);
617                 }
618         }
619         closelog();

621         (void) signal(SIGQUIT, SIG_DFL);
622         (void) signal(SIGINT, SIG_DFL);

624         /*
625          * Display some useful information to the new user like the banner
626          * and last login time if not a quiet login.
627          */

629         if (access(HUSHLOGIN, F_OK) != 0) {
630                 print_banner();
631                 display_last_login_time();
632         }
```

```
634         /*
635          * Set SIGXCPU and SIGXFSZ to default disposition.
636          * Shells inherit signal disposition from parent.
637          * And the shells should have default dispositions
638          * for the two below signals.
639          */
640         (void) signal(SIGXCPU, SIG_DFL);
641         (void) signal(SIGXFSZ, SIG_DFL);

643         /*
644          * Now fire off the shell of choice
645          */
646         exec_the_shell();

648         /*
649          * All done
650          */
651         login_exit(1);
652         return (0);
653 }


656 /*
657  *                      *** Utility functions ***
658  */



685 /* ONC_PLUS EXTRACT START */
662 /*
663  * donothing & catch    - Signal catching functions
664  */

666 /*ARGSUSED*/
667 static void
668 donothing(int sig)
669 {
670         if (pamh)
671                 (void) pam_end(pamh, PAM_ABORT);
672 }
697 /* ONC_PLUS EXTRACT END */

674 #ifdef notdef
675 static  int     intrupt;

677 /*ARGSUSED*/
678 static void
679 catch(int sig)
680 {
681         ++intrupt;
682 }
_____unchanged_portion_omitted_


818 /* ONC_PLUS EXTRACT START */
793 /*
794  * login_conv():
795  *      This is the conv (conversation) function called from
796  *      a PAM authentication module to print error messages
797  *      or garner information from the user.
798  */
799 /*ARGSUSED*/
800 static int
801 login_conv(int num_msg, struct pam_message **msg,
802     struct pam_response **response, void *appdata_ptr)
```



```
803 {
804         struct pam_message      *m;
805         struct pam_response     *r;
806         char                    *temp;
807         int                     k, i;

809         if (num_msg <= 0)
810                 return (PAM_CONV_ERR);

812         *response = calloc(num_msg, sizeof (struct pam_response));
813         if (*response == NULL)
814                 return (PAM_BUF_ERR);

816         k = num_msg;
817         m = *msg;
818         r = *response;
819         while (k--) {

821                 switch (m->msg_style) {

823                 case PAM_PROMPT_ECHO_OFF:
824                         errno = 0;
825                         temp = getpassphrase(m->msg);
826                         if (temp != NULL) {
827                                 if (errno == EINTR)
828                                         return (PAM_CONV_ERR);

830                                 r->resp = strdup(temp);
831                                 if (r->resp == NULL) {
832                                         /* free responses */
833                                         r = *response;
834                                         for (i = 0; i < num_msg; i++, r++) {
835                                                 if (r->resp)
836                                                         free(r->resp);
837                                         }
838                                         free(*response);
839                                         *response = NULL;
840                                         return (PAM_BUF_ERR);
841                                 }
842                         }

844                         m++;
845                         r++;
846                         break;

848                 case PAM_PROMPT_ECHO_ON:
849                         if (m->msg != NULL)
850                                 (void) fputs(m->msg, stdout);
851                         r->resp = calloc(1, PAM_MAX_RESP_SIZE);
852                         if (r->resp == NULL) {
853                                 /* free responses */
854                                 r = *response;
855                                 for (i = 0; i < num_msg; i++, r++) {
856                                         if (r->resp)
857                                                 free(r->resp);
858                                 }
859                                 free(*response);
860                                 *response = NULL;
861                                 return (PAM_BUF_ERR);
862                         }
863                         /*
864                          * The response might include environment variables
865                          * information. We should store that information in
866                          * envp if there is any; otherwise, envp is set to
867                          * NULL.
868                          */
```

```
                        bzero((void *)inputline, MAXLINE);

                        envp = getargs(inputline);

                        /* If we read in any input, process it. */
                        if (inputline[0] != '\0') {
                                int len;

                                if (envp != (char **)NULL)
                                        /*
                                         * If getargs() did not return NULL,
                                         * *envp is the first string in
                                         * inputline. envp++ makes envp point
                                         * to environment variables information
                                         *  or be NULL.
                                         */
                                        envp++;

                                (void) strncpy(r->resp, inputline,
                                    PAM_MAX_RESP_SIZE-1);
                                r->resp[PAM_MAX_RESP_SIZE-1] = NULL;
                                len = strlen(r->resp);
                                if (r->resp[len-1] == '\n')
                                        r->resp[len-1] = '\0';
                        } else {
                                login_exit(1);
                        }
                        m++;
                        r++;
                        break;

                case PAM_ERROR_MSG:
                        if (m->msg != NULL) {
                                (void) fputs(m->msg, stderr);
                                (void) fputs("\n", stderr);
                        }
                        m++;
                        r++;
                        break;
                case PAM_TEXT_INFO:
                        if (m->msg != NULL) {
                                (void) fputs(m->msg, stdout);
                                (void) fputs("\n", stdout);
                        }
                        m++;
                        r++;
                        break;

                default:
                        break;
                }
        }
        return (PAM_SUCCESS);
}
______unchanged_portion_omitted_
/* ONC_PLUS EXTRACT END */

/*
 * quotec               - Called by getargs
 */

static int
quotec(void)
{
        int c, i, num;
```



```
        switch (c = getc(stdin)) {

                case 'n':
                        c = '\n';
                        break;

                case 'r':
                        c = '\r';
                        break;

                case 'v':
                        c = '\013';
                        break;

                case 'b':
                        c = '\b';
                        break;

                case 't':
                        c = '\t';
                        break;

                case 'f':
                        c = '\f';
                        break;

                case '0':
                case '1':
                case '2':
                case '3':
                case '4':
                case '5':
                case '6':
                case '7':
                        for (num = 0, i = 0; i < 3; i++) {
                                num = num * 8 + (c - '0');
                                if ((c = getc(stdin)) < '0' || c > '7')
                                        break;
                        }
                        (void) ungetc(c, stdin);
                        c = num & 0377;
                        break;

                default:
                        break;
        }
        return (c);
}
______unchanged_portion_omitted_


/* ONC_PLUS EXTRACT START */
/*
 * login_authenticate   - Performs the main authentication work
 *                        1. Prints the login prompt
 *                        2. Requests and verifys the password
 *                        3. Checks the port password
 */

static void
login_authenticate(void)
{
        char *user;
        int err;
        int login_successful = 0;
```

```
1778         do {
1779                 /* if scheme broken, then nothing to do but quit */
1780                 if (pam_get_item(pamh, PAM_USER, (void **)&user) != PAM_SUCCESS)
1781                         exit(1);

1783                 /*
1784                  * only get name from utility if it is not already
1785                  * supplied by pam_start or a pam_set_item.
1786                  */
1787                 if (!user || !user[0]) {
1788                         /* use call back to get user name */
1789                         get_user_name();
1790                 }

1792                 err = verify_passwd();

1794                 /*
1795                  * If root login and not on system console then call exit(2)
1796                  */
1797                 check_for_console();

1799                 switch (err) {
1800                 case PAM_SUCCESS:
1801                 case PAM_NEW_AUTHTOK_REQD:
1802                         /*
1803                          * Officially, pam_authenticate() shouldn't return this
1804                          * but it's probably the right thing to return if
1805                          * PAM_DISALLOW_NULL_AUTHTOK is set so the user will
1806                          * be forced to change password later in this code.
1807                          */
1808                         count = 0;
1809                         login_successful = 1;
1810                         break;
1811                 case PAM_MAXTRIES:
1812                         count = retry;
1813                         /*FALLTHROUGH*/
1814                 case PAM_AUTH_ERR:
1815                 case PAM_AUTHINFO_UNAVAIL:
1816                 case PAM_USER_UNKNOWN:
1817                         audit_failure(get_audit_id(), ADT_FAIL_PAM + err, pwd,
1818                             remote_host, ttyn, zone_name);
1819                         log_bad_attempts();
1820                         break;
1821                 case PAM_ABORT:
1822                         log_bad_attempts();
1823                         (void) sleep(Disabletime);
1824                         (void) printf(incorrectmsg);

1826                         audit_error = ADT_FAIL_PAM + err;
1827                         login_exit(1);
1828                         /*NOTREACHED*/
1829                 default:        /* Some other PAM error */
1830                         audit_error = ADT_FAIL_PAM + err;
1831                         login_exit(1);
1832                         /*NOTREACHED*/
1833                 }

1835                 if (login_successful)
1836                         break;

1838                 /* sleep after bad passwd */
1839                 if (count)
1840                         (void) sleep(Sleeptime);
1841                 (void) printf(incorrectmsg);
1842                 /* force name to be null in this case */
1843                 if (pam_set_item(pamh, PAM_USER, NULL) != PAM_SUCCESS)
```



```
1844                         login_exit(1);
1845                 if (pam_set_item(pamh, PAM_RUSER, NULL) != PAM_SUCCESS)
1846                         login_exit(1);
1847         } while (count++ < retry);

1849         if (count >= retry) {
1850                 audit_failure(get_audit_id(), ADT_FAIL_VALUE_MAX_TRIES, pwd,
1851                     remote_host, ttyn, zone_name);
1852                 /*
1853                  * If logging is turned on, output the
1854                  * string storage area to the log file,
1855                  * and sleep for Disabletime
1856                  * seconds before exiting.
1857                  */
1858                 if (writelog)
1859                         badlogin();
1860                 if (dosyslog) {
1861                         if ((pwd = getpwnam(user_name)) != NULL) {
1862                                 if (remote_host[0]) {
1863                                         syslog(LOG_CRIT,
1864                                             "REPEATED LOGIN FAILURES ON %s "
1865                                             "FROM %.*s, %.*s",
1866                                             ttyn, HMAX, remote_host, NMAX,
1867                                             user_name);
1868                                 } else {
1869                                         syslog(LOG_CRIT,
1870                                             "REPEATED LOGIN FAILURES ON "
1871                                             "%s, %.*s",
1872                                             ttyn, NMAX, user_name);
1873                                 }
1874                         } else {
1875                                 if (remote_host[0]) {
1876                                         syslog(LOG_CRIT,
1877                                             "REPEATED LOGIN FAILURES ON %s "
1878                                             "FROM %.*s",
1879                                             ttyn, HMAX, remote_host);
1880                                 } else {
1881                                         syslog(LOG_CRIT,
1882                                             "REPEATED LOGIN FAILURES ON %s",
1883                                             ttyn);
1884                                 }
1885                         }
1886                 }
1887                 (void) sleep(Disabletime);
1888                 exit(1);
1889         }

1891 }
______unchanged_portion_omitted_
1970 /* ONC_PLUS EXTRACT END */

1943 static uint_t
1944 get_audit_id(void)
1945 {
1946         if (rflag)
1947                 return (ADT_rlogin);
1948         else if (hflag)
1949                 return (ADT_telnet);
1950         else if (zflag)
1951                 return (ADT_zlogin);

1953         return (ADT_login);
1954 }
______unchanged_portion_omitted_

2030 /* ONC_PLUS EXTRACT START */
```

```
2001 /*
2002  * update_utmpx_entry   - Searchs for the correct utmpx entry, making an
2003  *                        entry there if it finds one, otherwise exits.
2004  */

2006 static void
2007 update_utmpx_entry(int sublogin)
2008 {
2009         int     err;
2010         char    *user;
2011         static char     *errmsg = "No utmpx entry. "
2012             "You must exec \"login\" from the lowest level \"shell\".";
2013         int     tmplen;
2014         struct utmpx  *u = (struct utmpx *)0;
2015         struct utmpx  utmpx;
2016         char    *ttyntail;

2018         /*
2019          * If we're not a sublogin then
2020          * we'll get an error back if our PID doesn't match the PID of the
2021          * entry we are updating, otherwise if its a sublogin the flags
2022          * field is set to 0, which means we just write a matching entry
2023          * (without checking the pid), or a new entry if an entry doesn't
2024          * exist.
2025          */

2027         if ((err = pam_open_session(pamh, 0)) != PAM_SUCCESS) {
2028                 audit_error = ADT_FAIL_PAM + err;
2029                 login_exit(1);
2030         }

2032         if ((err = pam_get_item(pamh, PAM_USER, (void **) &user)) !=
2033             PAM_SUCCESS) {
2034                 audit_error = ADT_FAIL_PAM + err;
2035                 login_exit(1);
2036         }
2067 /* ONC_PLUS EXTRACT END */

2038         (void) memset((void *)&utmpx, 0, sizeof (utmpx));
2039         (void) time(&utmpx.ut_tv.tv_sec);
2040         utmpx.ut_pid = getpid();

2042         if (rflag || hflag) {
2043                 SCPYN(utmpx.ut_host, remote_host);
2044                 tmplen = strlen(remote_host) + 1;
2045                 if (tmplen < sizeof (utmpx.ut_host))
2046                         utmpx.ut_syslen = tmplen;
2047                 else
2048                         utmpx.ut_syslen = sizeof (utmpx.ut_host);
2049         } else if (zflag) {
2050                 /*
2051                  * If this is a login from another zone, put the
2052                  * zone:<zonename> string in the utmpx entry.
2053                  */
2054                 SCPYN(utmpx.ut_host, zone_name);
2055                 tmplen = strlen(zone_name) + 1;
2056                 if (tmplen < sizeof (utmpx.ut_host))
2057                         utmpx.ut_syslen = tmplen;
2058                 else
2059                         utmpx.ut_syslen = sizeof (utmpx.ut_host);
2060         } else {
2061                 utmpx.ut_syslen = 0;
2062         }

2064         SCPYN(utmpx.ut_user, user);
```



```
2066         /* skip over "/dev/" */
2067         ttyntail = basename(ttyn);

2069         while ((u = getutxent()) != NULL) {
2070                 if ((u->ut_type == INIT_PROCESS ||
2071                     u->ut_type == LOGIN_PROCESS ||
2072                     u->ut_type == USER_PROCESS) &&
2073                     ((sublogin && strncmp(u->ut_line, ttyntail,
2074                     sizeof (u->ut_line)) == 0) ||
2075                     u->ut_pid == login_pid)) {
2076                         SCPYN(utmpx.ut_line, (ttyn+sizeof ("/dev/")-1));
2077                         (void) memcpy(utmpx.ut_id, u->ut_id,
2078                             sizeof (utmpx.ut_id));
2079                         utmpx.ut_exit.e_exit = u->ut_exit.e_exit;
2080                         utmpx.ut_type = USER_PROCESS;
2081                         (void) pututxline(&utmpx);
2082                         break;
2083                 }
2084         }
2085         endutxent();

2087         if (u == (struct utmpx *)NULL) {
2088                 if (!sublogin) {
2089                         /*
2090                          * no utmpx entry already setup
2091                          * (init or rlogind/telnetd)
2092                          */
2093                         (void) puts(errmsg);

2095                         audit_error = ADT_FAIL_VALUE_PROGRAM;
2096                         login_exit(1);
2097                 }
2098         } else {
2099                 /* Now attempt to write out this entry to the wtmp file if */
2100                 /* we were successful in getting it from the utmpx file and */
2101                 /* the wtmp file exists.                                    */
2102                 updwtmpx(WTMPX_FILE, &utmpx);
2103         }
2135 /* ONC_PLUS EXTRACT START */
2104 }



2108 /*
2109  * process_chroot_logins        - Chroots to the specified subdirectory and
2110  *                                re executes login.
2111  */

2113 static int
2114 process_chroot_logins(void)
2115 {
2116         /*
2117          * If the shell field starts with a '*', do a chroot to the home
2118          * directory and perform a new login.
2119          */

2121         if (*pwd->pw_shell == '*') {
2122                 (void) pam_end(pamh, PAM_SUCCESS);      /* Done using PAM */
2123                 pamh = NULL;                            /* really done */
2124                 if (chroot(pwd->pw_dir) < 0) {
2125                         (void) printf("No Root Directory\n");

2127                         audit_failure(get_audit_id(),
2128                             ADT_FAIL_VALUE_CHDIR_FAILED,
2129                             pwd, remote_host, ttyn, zone_name);
```

```
2131                        return (ERROR);
2132                }
2133                /*
2134                 * Set the environment flag <!sublogin> so that the next login
2135                 * knows that it is a sublogin.
2136                 */
2169 /* ONC_PLUS EXTRACT END */
2137                envinit[0] = SUBLOGIN;
2138                envinit[1] = (char *)NULL;
2139                (void) printf("Subsystem root: %s\n", pwd->pw_dir);
2140                (void) execle("/usr/bin/login", "login", (char *)0,
2141                    &envinit[0]);
2142                (void) execle("/etc/login", "login", (char *)0, &envinit[0]);
2143                (void) printf("No /usr/bin/login or /etc/login on root\n");

2145                audit_error = ADT_FAIL_VALUE_PROGRAM;

2147                login_exit(1);
2148        }
2149        return (OK);
2183 /* ONC_PLUS EXTRACT START */
2150 }

2152 /*
2153  * establish_user_environment   - Set up the new users enviornment
2154  */

2156 static void
2157 establish_user_environment(char **renvp)
2158 {
2159        int i, j, k, l_index, length, idx = 0;
2160        char *endptr;
2161        char **lenvp;
2162        char **pam_env;

2164        lenvp = environ;
2165        while (*lenvp++)
2166                ;

2168        /* count the number of PAM environment variables set by modules */
2169        if ((pam_env = pam_getenvlist(pamh)) != 0) {
2170                for (idx = 0; pam_env[idx] != 0; idx++)
2171                        ;
2172        }

2174        envinit = (char **)calloc(lenvp - environ + 10 + MAXARGS + idx,
2175            sizeof (char *));
2176        if (envinit == NULL) {
2177                (void) printf("Calloc failed - out of swap space.\n");
2178                login_exit(8);
2179        }

2181        /*
2182         * add PAM environment variables first so they
2183         * can be overwritten at login's discretion.
2184         * check for illegal environment variables.
2185         */
2186        idx = 0;        basicenv = 0;
2187        if (pam_env != 0) {
2188                while (pam_env[idx] != 0) {
2189                        if (legalenvvar(pam_env[idx])) {
2190                                envinit[basicenv] = pam_env[idx];
2191                                basicenv++;
2192                        }
2193                        idx++;
2194                }
```



```
2195        }
2196        (void) memcpy(&envinit[basicenv], newenv, sizeof (newenv));
2231 /* ONC_PLUS EXTRACT END */

2198        /* Set up environment */
2199        if (rflag) {
2200                ENVSTRNCAT(term, terminal);
2201        } else if (hflag) {
2202                if (strlen(terminal)) {
2203                        ENVSTRNCAT(term, terminal);
2204                }
2205        } else {
2206                char *tp = getenv("TERM");

2208                if ((tp != NULL) && (*tp != '\0'))
2209                        ENVSTRNCAT(term, tp);
2210        }

2212        ENVSTRNCAT(logname, pwd->pw_name);

2214        /*
2215         * There are three places to get timezone info.  init.c sets
2216         * TZ if the file /etc/TIMEZONE contains a value for TZ.
2217         * login.c looks in the file /etc/default/login for a
2218         * variable called TIMEZONE being set.  If TIMEZONE has a
2219         *  value, TZ is set to that value; no environment variable
2220         * TIMEZONE is set, only TZ.  If neither of these methods
2221         * work to set TZ, then the library routines  will default
2222         * to using the file /usr/lib/locale/TZ/localtime.
2223         *
2224         * There is a priority set up here.  If /etc/TIMEZONE has
2225         * a value for TZ, that value remains top priority.  If the
2226         * file /etc/default/login has TIMEZONE set, that has second
2227         * highest priority not overriding the value of TZ in
2228         * /etc/TIMEZONE.  The reason for this priority is that the
2229         * file /etc/TIMEZONE is supposed to be sourced by
2230         * /etc/profile.  We are doing the "sourcing" prematurely in
2231         * init.c.  Additionally, a login C shell doesn't source the
2232         * file /etc/profile thus not sourcing /etc/TIMEZONE thus not
2233         * allowing an adminstrator to globally set TZ for all users
2234         */
2235        if (Def_tz != NULL)     /* Is there a TZ from defaults/login? */
2236                tmp_tz = Def_tz;

2238        if ((Def_tz = getenv("TZ")) != NULL) {
2239                ENVSTRNCAT(timez, Def_tz);
2240        } else if (tmp_tz != NULL) {
2241                Def_tz = tmp_tz;
2242                ENVSTRNCAT(timez, Def_tz);
2243        }

2245        if (Def_hertz == NULL)
2246                (void) sprintf(hertz + strlen(hertz), "%lu", HZ);
2247        else
2248                ENVSTRNCAT(hertz, Def_hertz);

2250        if (Def_path == NULL)
2251                (void) strlcat(path, DEF_PATH, sizeof (path));
2252        else
2253                ENVSTRNCAT(path, Def_path);

2255        ENVSTRNCAT(home, pwd->pw_dir);

2257        /*
2258         * Find the end of the basic environment
2259         */
```

```
2260         for (basicenv = 0; envinit[basicenv] != NULL; basicenv++)
2261                 ;

2263         /*
2264          * If TZ has a value, add it.
2265          */
2266         if (strcmp(timez, "TZ=") != 0)
2267                 envinit[basicenv++] = timez;

2269         if (*pwd->pw_shell == '\0') {
2270                 /*
2271                  * If possible, use the primary default shell,
2272                  * otherwise, use the secondary one.
2273                  */
2274                 if (access(SHELL, X_OK) == 0)
2275                         pwd->pw_shell = SHELL;
2276                 else
2277                         pwd->pw_shell = SHELL2;
2278         } else if (Altshell != NULL && strcmp(Altshell, "YES") == 0) {
2279                 envinit[basicenv++] = shell;
2280                 ENVSTRNCAT(shell, pwd->pw_shell);
2281         }

2283 #ifndef NO_MAIL
2284         envinit[basicenv++] = mail;
2285         (void) strlcat(mail, pwd->pw_name, sizeof (mail));
2286 #endif

2288         /*
2289          * Pick up locale environment variables, if any.
2290          */
2291         lenvp = renvp;
2292         while (*lenvp != NULL) {
2293                 j = 0;
2294                 while (localeenv[j] != 0) {
2295                         /*
2296                          * locale_envmatch() returns 1 if
2297                          * *lenvp is localenev[j] and valid.
2298                          */
2299                         if (locale_envmatch(localeenv[j], *lenvp) == 1) {
2300                                 envinit[basicenv++] = *lenvp;
2301                                 break;
2302                         }
2303                         j++;
2304                 }
2305                 lenvp++;
2306         }

2308         /*
2309          * If '-p' flag, then try to pass on allowable environment
2310          * variables.  Note that by processing this first, what is
2311          * passed on the final "login:" line may over-ride the invocation
2312          * values.  XXX is this correct?
2313          */
2314         if (pflag) {
2315                 for (lenvp = renvp; *lenvp; lenvp++) {
2316                         if (!legalenvvar(*lenvp)) {
2317                                 continue;
2318                         }
2319                         /*
2320                          * If this isn't 'xxx=yyy', skip it.  XXX
2321                          */
2322                         if ((endptr = strchr(*lenvp, '=')) == NULL) {
2323                                 continue;
2324                         }
2325                         length = endptr + 1 - *lenvp;
```



```
2326                         for (j = 0; j < basicenv; j++) {
2327                                 if (strncmp(envinit[j], *lenvp, length) == 0) {
2328                                         /*
2329                                          * Replace previously established value
2330                                          */
2331                                         envinit[j] = *lenvp;
2332                                         break;
2333                                 }
2334                         }
2335                         if (j == basicenv) {
2336                                 /*
2337                                  * It's a new definition, so add it at the end.
2338                                  */
2339                                 envinit[basicenv++] = *lenvp;
2340                         }
2341                 }
2342         }

2344         /*
2345          * Add in all the environment variables picked up from the
2346          * argument list to "login" or from the user response to the
2347          * "login" request, if any.
2348          */

2350         if (envp == NULL)
2351                 goto switch_env;        /* done */

2353         for (j = 0, k = 0, l_index = 0;
2354             *envp != NULL && j < (MAXARGS-1);
2355             j++, envp++) {

2357                 /*
2358                  * Scan each string provided.  If it doesn't have the
2359                  * format xxx=yyy, then add the string "Ln=" to the beginning.
2360                  */
2361                 if ((endptr = strchr(*envp, '=')) == NULL) {
2362                         /*
2363                          * This much to be malloc'd:
2364                          *   strlen(*envp) + 1 char for 'L' +
2365                          *   MAXARGSWIDTH + 1 char for '=' + 1 for null char;
2366                          *
2367                          * total = strlen(*envp) + MAXARGSWIDTH + 3
2368                          */
2369                         int total = strlen(*envp) + MAXARGSWIDTH + 3;
2370                         envinit[basicenv+k] = malloc(total);
2371                         if (envinit[basicenv+k] == NULL) {
2372                                 (void) printf("%s: malloc failed\n", PROG_NAME);
2373                                 login_exit(1);
2374                         }
2375                         (void) snprintf(envinit[basicenv+k], total, "L%d=%s",
2376                             l_index, *envp);

2378                         k++;
2379                         l_index++;
2380                 } else  {
2381                         if (!legalenvvar(*envp)) { /* this env var permited? */
2382                                 continue;
2383                         } else {

2385                                 /*
2386                                  * Check to see whether this string replaces
2387                                  * any previously defined string
2388                                  */
2389                                 for (i = 0, length = endptr + 1 - *envp;
2390                                     i < basicenv + k; i++) {
2391                                         if (strncmp(*envp, envinit[i], length)
```

```
2392                                            == 0) {
2393                                                envinit[i] = *envp;
2394                                                break;
2395                                        }
2396                                }

2398                                /*
2399                                 * If it doesn't, place it at the end of
2400                                 * environment array.
2401                                 */
2402                                if (i == basicenv+k) {
2403                                        envinit[basicenv+k] = *envp;
2404                                        k++;
2405                                }
2406                        }
2407                }
2408        }               /* for (j = 0 ... ) */

2410 switch_env:
2411         /*
2412          * Switch to the new environment.
2413          */
2414         environ = envinit;
2415 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2408 Thu Jul 11 01:28:57 2013
new/usr/src/cmd/sendmail/src/Makefile
first pass
**********************************************************
#
# 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) 2011 Gary Mills

#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# cmd/sendmail/src/Makefile
#

PROG=   sendmail

include         ../../Makefile.cmd
include         ../Makefile.cmd

OBJS= alias.o arpadate.o bf.o collect.o conf.o control.o convtime.o daemon.o \
        deliver.o domain.o envelope.o err.o headers.o macro.o main.o map.o \
        mci.o milter.o mime.o parseaddr.o queue.o ratectrl.o readcf.o \
        recipient.o sasl.o savemail.o sfsasl.o sm_resolve.o srvrsmtp.o stab.o \
        stats.o sysexits.o tls.o trace.o udb.o usersmtp.o util.o version.o

SRCS=   $(OBJS:%.o=%.c)

MAPFILES =      $(MAPFILE.INT) $(MAPFILE.NGB)
LDFLAGS +=      $(MAPFILES:%=-M%)

# EXPORT DELETE START
CRYPTOLIBS=     -lssl -lcrypto -lsasl
# EXPORT DELETE END
LDLIBS +=       ../libsmutil/libsmutil.a ../libsm/libsm.a -lresolv -lsocket \
                -lnsl ../db/libdb.a -lldap -lsldap -lwrap -lumem \
                -lssl -lcrypto -lsasl
                $(CRYPTOLIBS)

INCPATH=        -I. -I../include -I../db

ENVDEF=         -DNETINET6 -DTCPWRAPPERS -DSTARTTLS -DSASL=20115
# EXPORT DELETE START
CRYPTOENVDEF=   -DSTARTTLS -DSASL=20115
# EXPORT DELETE END
ENVDEF=         -DNETINET6 -DTCPWRAPPERS $(CRYPTOENVDEF)
```



```
SUNENVDEF=      -DSUN_EXTENSIONS -DVENDOR_DEFAULT=VENDOR_SUN \
                -DSUN_INIT_DOMAIN -DSUN_SIMPLIFIED_LDAP -D_FFR_LOCAL_DAEMON \
                -D_FFR_MAIL_MACRO

CPPFLAGS =      $(INCPATH) $(ENVDEF) $(SUNENVDEF) $(DBMDEF) $(CPPFLAGS.sm)

FILEMODE=       2555

ROOTSYMLINKS=   $(ROOTUSRSBIN)/newaliases $(ROOTUSRSBIN)/sendmail

# build rule
#

.KEEP_STATE:
all:            $(PROG)

.PARALLEL:      $(OBJS)

$(PROG):        $(OBJS) $(MAPFILES) \
        ../libsmutil/libsmutil.a ../libsm/libsm.a ../db/libdb.a
        $(LINK.c) -o $@ $(OBJS) $(LDLIBS)
        $(POST_PROCESS)

install:        $(ROOTLIBPROG) $(ROOTSYMLINKS)

$(ROOTSYMLINKS):
        $(RM) $@; $(SYMLINK) ../lib/sendmail $@

clean:
        $(RM) $(PROG) $(OBJS)

lint:   lint_SRCS

# EXPORT DELETE START
EXPORT_SRC:
        $(RM) Makefile+
        $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
            < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile
# EXPORT DELETE END

include         ../../Makefile.targ
```

```
**********************************************************
    1011 Thu Jul 11 01:28:58 2013
new/usr/src/common/crypto/aes/Makefile
first pass
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #
  25 # common/crypto/aes/Makefile
  26 #
  27 # include global definitions
  28 include $(SRC)/Makefile.master

  30 .KEEP_STATE:

  32 FRC:


  35 # EXPORT DELETE START
  36 EXPORT_SRC:
  37         $(RM) Makefile+ aes_impl.c+ aes_impl.h+ amd64/aes_amd64.s+ \
  38                 amd64/aes_intel.s+ sun4u/aes_crypt_asm.s+
  39         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  40                 < aes_impl.c > aes_impl.c+
  41         $(MV) aes_impl.c+ aes_impl.c
  42         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  43                 < aes_impl.h > aes_impl.h+
  44         $(MV) aes_impl.h+ aes_impl.h
  45         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  46                 < amd64/aes_amd64.s > amd64/aes_amd64.s+
  47         $(MV) amd64/aes_amd64.s+ amd64/aes_amd64.s
  48         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  49                 < amd64/aes_intel.s > amd64/aes_intel.s+
  50         $(MV) amd64/aes_intel.s+ amd64/aes_intel.s
  51         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  52                 < sun4u/aes_crypt_asm.s > sun4u/aes_crypt_asm.s+
  53         $(MV) sun4u/aes_crypt_asm.s+ sun4u/aes_crypt_asm.s
  54         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
  55                 < Makefile > Makefile+
  56         $(RM) Makefile
  57         $(MV) Makefile+ Makefile
  58         $(CHMOD) 444 Makefile aes_impl.c aes_impl.h amd64/aes_amd64.s \
  59                 amd64/aes_intel.s sun4u/aes_crypt_asm.s

  61 # EXPORT DELETE END
```

```
**********************************************************
   62998 Thu Jul 11 01:28:58 2013
new/usr/src/common/crypto/aes/aes_impl.c
first pass
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright (c) 2003, 2010, Oracle and/or its affiliates. All rights reserved.
  23  */

  25 #include <sys/types.h>
  26 #include <sys/systm.h>
  27 #include <sys/sysmacros.h>
  28 #include <netinet/in.h>
  29 #include "aes_impl.h"
  30 #ifndef _KERNEL
  31 #include <strings.h>
  32 #include <stdlib.h>
  33 #endif  /* !_KERNEL */

  35 #ifdef __amd64

  37 #ifdef _KERNEL
  38 #include <sys/cpuvar.h>         /* cpu_t, CPU */
  39 #include <sys/x86_archext.h>    /* x86_featureset, X86FSET_AES */
  40 #include <sys/disp.h>           /* kpreempt_disable(), kpreempt_enable */

  42 /* Workaround for no XMM kernel thread save/restore */
  43 #define KPREEMPT_DISABLE        kpreempt_disable()
  44 #define KPREEMPT_ENABLE         kpreempt_enable()

  46 #else
  47 #include <sys/auxv.h>           /* getisax() */
  48 #include <sys/auxv_386.h>       /* AV_386_AES bit */
  49 #define KPREEMPT_DISABLE
  50 #define KPREEMPT_ENABLE
  51 #endif  /* _KERNEL */
  52 #endif  /* __amd64 */


  55 /*
  56  * This file is derived from the file  rijndael-alg-fst.c  taken from the
  57  * "optimized C code v3.0" on the "rijndael home page"
  58  * http://www.iaik.tu-graz.ac.at/research/krypto/AES/old/~rijmen/rijndael/
  59  * pointed by the NIST web-site http://csrc.nist.gov/archive/aes/
  60  *
  61  * The following note is from the original file:
```



```
  62  */

  64 /*
  65  * rijndael-alg-fst.c
  66  *
  67  * @version 3.0 (December 2000)
  68  *
  69  * Optimised ANSI C code for the Rijndael cipher (now AES)
  70  *
  71  * @author Vincent Rijmen <vincent.rijmen@esat.kuleuven.ac.be>
  72  * @author Antoon Bosselaers <antoon.bosselaers@esat.kuleuven.ac.be>
  73  * @author Paulo Barreto <paulo.barreto@terra.com.br>
  74  *
  75  * This code is hereby placed in the public domain.
  76  *
  77  * THIS SOFTWARE IS PROVIDED BY THE AUTHORS ''AS IS'' AND ANY EXPRESS
  78  * OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
  79  * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
  80  * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE
  81  * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
  82  * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
  83  * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
  84  * BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
  85  * WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
  86  * OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
  87  * EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
  88  */

  90 /* EXPORT DELETE START */

  90 #if defined(sun4u)
  91 /* External assembly functions: */
  92 extern void aes_encrypt_impl(const uint32_t rk[], int Nr, const uint32_t pt[4],
  93         uint32_t ct[4]);
  94 extern void aes_decrypt_impl(const uint32_t rk[], int Nr, const uint32_t ct[4],
  95         uint32_t pt[4]);

  97 #define AES_ENCRYPT_IMPL(a, b, c, d, e) aes_encrypt_impl(a, b, c, d)
  98 #define AES_DECRYPT_IMPL(a, b, c, d, e) aes_decrypt_impl(a, b, c, d)

 100 #elif defined(__amd64)

 102 /* These functions are used to execute amd64 instructions for AMD or Intel: */
 103 extern int rijndael_key_setup_enc_amd64(uint32_t rk[],
 104         const uint32_t cipherKey[], int keyBits);
 105 extern int rijndael_key_setup_dec_amd64(uint32_t rk[],
 106         const uint32_t cipherKey[], int keyBits);
 107 extern void aes_encrypt_amd64(const uint32_t rk[], int Nr,
 108         const uint32_t pt[4], uint32_t ct[4]);
 109 extern void aes_decrypt_amd64(const uint32_t rk[], int Nr,
 110         const uint32_t ct[4], uint32_t pt[4]);

 112 /* These functions are used to execute Intel-specific AES-NI instructions: */
 113 extern int rijndael_key_setup_enc_intel(uint32_t rk[],
 114         const uint32_t cipherKey[], uint64_t keyBits);
 115 extern int rijndael_key_setup_dec_intel(uint32_t rk[],
 116         const uint32_t cipherKey[], uint64_t keyBits);
 117 extern void aes_encrypt_intel(const uint32_t rk[], int Nr,
 118         const uint32_t pt[4], uint32_t ct[4]);
 119 extern void aes_decrypt_intel(const uint32_t rk[], int Nr,
 120         const uint32_t ct[4], uint32_t pt[4]);

 122 static int intel_aes_instructions_present(void);

 124 #define AES_ENCRYPT_IMPL(a, b, c, d, e) rijndael_encrypt(a, b, c, d, e)
 125 #define AES_DECRYPT_IMPL(a, b, c, d, e) rijndael_decrypt(a, b, c, d, e)
```

```
 127 #else /* Generic C implementation */

 129 #define AES_ENCRYPT_IMPL(a, b, c, d, e) rijndael_encrypt(a, b, c, d)
 130 #define AES_DECRYPT_IMPL(a, b, c, d, e) rijndael_decrypt(a, b, c, d)
 131 #define rijndael_key_setup_enc_raw      rijndael_key_setup_enc
 132 #endif  /* sun4u || __amd64 */

 134 #if defined(_LITTLE_ENDIAN) && !defined(__amd64)
 135 #define AES_BYTE_SWAP
 136 #endif


 139 #if !defined(__amd64)
 140 /*
 141  *  Constant tables
 142  */

 144 /*
 145  * Te0[x] = S [x].[02, 01, 01, 03];
 146  * Te1[x] = S [x].[03, 02, 01, 01];
 147  * Te2[x] = S [x].[01, 03, 02, 01];
 148  * Te3[x] = S [x].[01, 01, 03, 02];
 149  * Te4[x] = S [x].[01, 01, 01, 01];
 150  *
 151  * Td0[x] = Si[x].[0e, 09, 0d, 0b];
 152  * Td1[x] = Si[x].[0b, 0e, 09, 0d];
 153  * Td2[x] = Si[x].[0d, 0b, 0e, 09];
 154  * Td3[x] = Si[x].[09, 0d, 0b, 0e];
 155  * Td4[x] = Si[x].[01, 01, 01, 01];
 156  */

 158 /* Encrypt Sbox constants (for the substitute bytes operation) */

 160 #ifndef sun4u

 162 static const uint32_t Te0[256] =
 163 {
 164         0xc66363a5U, 0xf87c7c84U, 0xee777799U, 0xf67b7b8dU,
 165         0xfff2f20dU, 0xd66b6bbdU, 0xde6f6fb1U, 0x91c5c554U,
 166         0x60303050U, 0x02010103U, 0xce6767a9U, 0x562b2b7dU,
 167         0xe7fefe19U, 0xb5d7d762U, 0x4dababe6U, 0xec76769aU,
 168         0x8fcaca45U, 0x1f82829dU, 0x89c9c940U, 0xfa7d7d87U,
 169         0xeffafa15U, 0xb25959ebU, 0x8e4747c9U, 0xfbf0f00bU,
 170         0x41adadecU, 0xb3d4d467U, 0x5fa2a2fdU, 0x45afafeaU,
 171         0x239c9cbfU, 0x53a4a4f7U, 0xe4727296U, 0x9bc0c05bU,
 172         0x75b7b7c2U, 0xe1fdfd1cU, 0x3d9393aeU, 0x4c26266aU,
 173         0x6c36365aU, 0x7e3f3f41U, 0xf5f7f702U, 0x83cccc4fU,
 174         0x6834345cU, 0x51a5a5f4U, 0xd1e5e534U, 0xf9f1f108U,
 175         0xe2717193U, 0xabd8d873U, 0x62313153U, 0x2a15153fU,
 176         0x0804040cU, 0x95c7c752U, 0x46232365U, 0x9dc3c35eU,
 177         0x30181828U, 0x379696a1U, 0x0a05050fU, 0x2f9a9ab5U,
 178         0x0e070709U, 0x24121236U, 0x1b80809bU, 0xdfe2e23dU,
 179         0xcdebeb26U, 0x4e272769U, 0x7fb2b2cdU, 0xea75759fU,
 180         0x1209091bU, 0x1d83839eU, 0x582c2c74U, 0x341a1a2eU,
 181         0x361b1b2dU, 0xdc6e6eb2U, 0xb45a5aeeU, 0x5ba0a0fbU,
 182         0xa45252f6U, 0x763b3b4dU, 0xb7d6d661U, 0x7db3b3ceU,
 183         0x5229297bU, 0xdde3e33eU, 0x5e2f2f71U, 0x13848497U,
 184         0xa65353f5U, 0xb9d1d168U, 0x00000000U, 0xc1eded2cU,
 185         0x40202060U, 0xe3fcfc1fU, 0x79b1b1c8U, 0xb65b5bedU,
 186         0xd46a6abeU, 0x8dcbcb46U, 0x67bebed9U, 0x7239394bU,
 187         0x944a4adeU, 0x984c4cd4U, 0xb05858e8U, 0x85cfcf4aU,
 188         0xbbd0d06bU, 0xc5efef2aU, 0x4faaaae5U, 0xedfbfb16U,
 189         0x864343c5U, 0x9a4d4dd7U, 0x66333355U, 0x11858594U,
 190         0x8a4545cfU, 0xe9f9f910U, 0x04020206U, 0xfe7f7f81U,
 191         0xa05050f0U, 0x783c3c44U, 0x259f9fbaU, 0x4ba8a8e3U,
```



```
 192         0xa25151f3U, 0x5da3a3feU, 0x804040c0U, 0x058f8f8aU,
 193         0x3f9292adU, 0x219d9dbcU, 0x70383848U, 0xf1f5f504U,
 194         0x63bcbcdfU, 0x77b6b6c1U, 0xafdada75U, 0x42212163U,
 195         0x20101030U, 0xe5ffff1aU, 0xfdf3f30eU, 0xbfd2d26dU,
 196         0x81cdcd4cU, 0x180c0c14U, 0x26131335U, 0xc3ecec2fU,
 197         0xbe5f5fe1U, 0x359797a2U, 0x884444ccU, 0x2e171739U,
 198         0x93c4c457U, 0x55a7a7f2U, 0xfc7e7e82U, 0x7a3d3d47U,
 199         0xc86464acU, 0xba5d5de7U, 0x3219192bU, 0xe6737395U,
 200         0xc06060a0U, 0x19818198U, 0x9e4f4fd1U, 0xa3dcdc7fU,
 201         0x44222266U, 0x542a2a7eU, 0x3b9090abU, 0x0b888883U,
 202         0x8c4646caU, 0xc7eeee29U, 0x6bb8b8d3U, 0x2814143cU,
 203         0xa7dede79U, 0xbc5e5ee2U, 0x160b0b1dU, 0xaddbdb76U,
 204         0xdbe0e03bU, 0x64323256U, 0x743a3a4eU, 0x140a0a1eU,
 205         0x924949dbU, 0x0c06060aU, 0x4824246cU, 0xb85c5ce4U,
 206         0x9fc2c25dU, 0xbdd3d36eU, 0x43acacefU, 0xc46262a6U,
 207         0x399191a8U, 0x319595a4U, 0xd3e4e437U, 0xf279798bU,
 208         0xd5e7e732U, 0x8bc8c843U, 0x6e373759U, 0xda6d6db7U,
 209         0x018d8d8cU, 0xb1d5d564U, 0x9c4e4ed2U, 0x49a9a9e0U,
 210         0xd86c6cb4U, 0xac5656faU, 0xf3f4f407U, 0xcfeaea25U,
 211         0xca6565afU, 0xf47a7a8eU, 0x47aeaee9U, 0x10080818U,
 212         0x6fbabad5U, 0xf0787888U, 0x4a25256fU, 0x5c2e2e72U,
 213         0x381c1c24U, 0x57a6a6f1U, 0x73b4b4c7U, 0x97c6c651U,
 214         0xcbe8e823U, 0xa1dddd7cU, 0xe874749cU, 0x3e1f1f21U,
 215         0x964b4bddU, 0x61bdbddcU, 0x0d8b8b86U, 0x0f8a8a85U,
 216         0xe0707090U, 0x7c3e3e42U, 0x71b5b5c4U, 0xcc6666aaU,
 217         0x904848d8U, 0x06030305U, 0xf7f6f601U, 0x1c0e0e12U,
 218         0xc26161a3U, 0x6a35355fU, 0xae5757f9U, 0x69b9b9d0U,
 219         0x17868691U, 0x99c1c158U, 0x3a1d1d27U, 0x279e9eb9U,
 220         0xd9e1e138U, 0xebf8f813U, 0x2b9898b3U, 0x22111133U,
 221         0xd26969bbU, 0xa9d9d970U, 0x078e8e89U, 0x339494a7U,
 222         0x2d9b9bb6U, 0x3c1e1e22U, 0x15878792U, 0xc9e9e920U,
 223         0x87cece49U, 0xaa5555ffU, 0x50282878U, 0xa5dfdf7aU,
 224         0x038c8c8fU, 0x59a1a1f8U, 0x09898980U, 0x1a0d0d17U,
 225         0x65bfbfdaU, 0xd7e6e631U, 0x844242c6U, 0xd06868b8U,
 226         0x824141c3U, 0x299999b0U, 0x5a2d2d77U, 0x1e0f0f11U,
 227         0x7bb0b0cbU, 0xa85454fcU, 0x6dbbbbd6U, 0x2c16163aU
 228 };
_____unchanged_portion_omitted_
1556 #endif  /* sun4u, __amd64 */
1559 /* EXPORT DELETE END */


1559 /*
1560  * Initialize AES encryption and decryption key schedules.
1561  *
1562  * Parameters:
1563  * cipherKey    User key
1564  * keyBits      AES key size (128, 192, or 256 bits)
1565  * keysched     AES key schedule to be initialized, of type aes_key_t.
1566  *              Allocated by aes_alloc_keysched().
1567  */
1568 void
1569 aes_init_keysched(const uint8_t *cipherKey, uint_t keyBits, void *keysched)
1570 {
1574 /* EXPORT DELETE START */
1571         aes_key_t       *newbie = keysched;
1572         uint_t          keysize, i, j;
1573         union {
1574                 uint64_t        ka64[4];
1575                 uint32_t        ka32[8];
1576                 } keyarr;

1578         switch (keyBits) {
1579         case 128:
1580                 newbie->nr = 10;
1581                 break;
```

```
1583         case 192:
1584                 newbie->nr = 12;
1585                 break;

1587         case 256:
1588                 newbie->nr = 14;
1589                 break;

1591         default:
1592                 /* should never get here */
1593                 return;
1594         }
1595         keysize = CRYPTO_BITS2BYTES(keyBits);

1597         /*
1598          * For _LITTLE_ENDIAN machines (except AMD64), reverse every
1599          * 4 bytes in the key.  On _BIG_ENDIAN and AMD64, copy the key
1600          * without reversing bytes.
1601          * For AMD64, do not byte swap for aes_setupkeys().
1602          *
1603          * SPARCv8/v9 uses a key schedule array with 64-bit elements.
1604          * X86/AMD64  uses a key schedule array with 32-bit elements.
1605          */
1606 #ifndef AES_BYTE_SWAP
1607         if (IS_P2ALIGNED(cipherKey, sizeof (uint64_t))) {
1608                 for (i = 0, j = 0; j < keysize; i++, j += 8) {
1609                         /* LINTED: pointer alignment */
1610                         keyarr.ka64[i] = *((uint64_t *)&cipherKey[j]);
1611                 }
1612         } else {
1613                 bcopy(cipherKey, keyarr.ka32, keysize);
1614         }

1616 #else   /* byte swap */
1617         for (i = 0, j = 0; j < keysize; i++, j += 4) {
1618                 keyarr.ka32[i] = htonl(*(uint32_t *)(void *)&cipherKey[j]);
1619         }
1620 #endif

1622         aes_setupkeys(newbie, keyarr.ka32, keyBits);
1627 /* EXPORT DELETE END */
1623 }


1626 /*
1627  * Encrypt one block using AES.
1628  * Align if needed and (for x86 32-bit only) byte-swap.
1629  *
1630  * Parameters:
1631  * ks   Key schedule, of type aes_key_t
1632  * pt   Input block (plain text)
1633  * ct   Output block (crypto text).  Can overlap with pt
1634  */
1635 int
1636 aes_encrypt_block(const void *ks, const uint8_t *pt, uint8_t *ct)
1637 {
1643 /* EXPORT DELETE START */
1638         aes_key_t       *ksch = (aes_key_t *)ks;

1640 #ifndef AES_BYTE_SWAP
1641         if (IS_P2ALIGNED2(pt, ct, sizeof (uint32_t))) {
1642                 /* LINTED:  pointer alignment */
1643                 AES_ENCRYPT_IMPL(&ksch->encr_ks.ks32[0], ksch->nr,
1644                     /* LINTED:  pointer alignment */
1645                     (uint32_t *)pt, (uint32_t *)ct, ksch->flags);
```



```
1646         } else {
1647 #endif
1648                 uint32_t buffer[AES_BLOCK_LEN / sizeof (uint32_t)];

1650                 /* Copy input block into buffer */
1651 #ifndef AES_BYTE_SWAP
1652                 bcopy(pt, &buffer, AES_BLOCK_LEN);

1654 #else   /* byte swap */
1655                 buffer[0] = htonl(*(uint32_t *)(void *)&pt[0]);
1656                 buffer[1] = htonl(*(uint32_t *)(void *)&pt[4]);
1657                 buffer[2] = htonl(*(uint32_t *)(void *)&pt[8]);
1658                 buffer[3] = htonl(*(uint32_t *)(void *)&pt[12]);
1659 #endif

1661                 AES_ENCRYPT_IMPL(&ksch->encr_ks.ks32[0], ksch->nr,
1662                     buffer, buffer, ksch->flags);

1664                 /* Copy result from buffer to output block */
1665 #ifndef AES_BYTE_SWAP
1666                 bcopy(&buffer, ct, AES_BLOCK_LEN);
1667         }

1669 #else   /* byte swap */
1670                 *(uint32_t *)(void *)&ct[0] = htonl(buffer[0]);
1671                 *(uint32_t *)(void *)&ct[4] = htonl(buffer[1]);
1672                 *(uint32_t *)(void *)&ct[8] = htonl(buffer[2]);
1673                 *(uint32_t *)(void *)&ct[12] = htonl(buffer[3]);
1674 #endif
1681 /* EXPORT DELETE END */
1675         return (CRYPTO_SUCCESS);
1676 }


1679 /*
1680  * Decrypt one block using AES.
1681  * Align and byte-swap if needed.
1682  *
1683  * Parameters:
1684  * ks   Key schedule, of type aes_key_t
1685  * ct   Input block (crypto text)
1686  * pt   Output block (plain text). Can overlap with pt
1687  */
1688 int
1689 aes_decrypt_block(const void *ks, const uint8_t *ct, uint8_t *pt)
1690 {
1698 /* EXPORT DELETE START */
1691         aes_key_t       *ksch = (aes_key_t *)ks;

1693 #ifndef AES_BYTE_SWAP
1694         if (IS_P2ALIGNED2(ct, pt, sizeof (uint32_t))) {
1695                 /* LINTED:  pointer alignment */
1696                 AES_DECRYPT_IMPL(&ksch->decr_ks.ks32[0], ksch->nr,
1697                     /* LINTED:  pointer alignment */
1698                     (uint32_t *)ct, (uint32_t *)pt, ksch->flags);
1699         } else {
1700 #endif
1701                 uint32_t buffer[AES_BLOCK_LEN / sizeof (uint32_t)];

1703                 /* Copy input block into buffer */
1704 #ifndef AES_BYTE_SWAP
1705                 bcopy(ct, &buffer, AES_BLOCK_LEN);

1707 #else   /* byte swap */
1708                 buffer[0] = htonl(*(uint32_t *)(void *)&ct[0]);
1709                 buffer[1] = htonl(*(uint32_t *)(void *)&ct[4]);
```

```
1710                 buffer[2] = htonl(*(uint32_t *)(void *)&ct[8]);
1711                 buffer[3] = htonl(*(uint32_t *)(void *)&ct[12]);
1712 #endif

1714                 AES_DECRYPT_IMPL(&ksch->decr_ks.ks32[0], ksch->nr,
1715                     buffer, buffer, ksch->flags);

1717                 /* Copy result from buffer to output block */
1718 #ifndef AES_BYTE_SWAP
1719                 bcopy(&buffer, pt, AES_BLOCK_LEN);
1720         }

1722 #else   /* byte swap */
1723         *(uint32_t *)(void *)&pt[0] = htonl(buffer[0]);
1724         *(uint32_t *)(void *)&pt[4] = htonl(buffer[1]);
1725         *(uint32_t *)(void *)&pt[8] = htonl(buffer[2]);
1726         *(uint32_t *)(void *)&pt[12] = htonl(buffer[3]);
1727 #endif

1737 /* EXPORT DELETE END */
1729         return (CRYPTO_SUCCESS);
1730 }


1733 /*
1734  * Allocate key schedule for AES.
1735  *
1736  * Return the pointer and set size to the number of bytes allocated.
1737  * Memory allocated must be freed by the caller when done.
1738  *
1739  * Parameters:
1740  * size         Size of key schedule allocated, in bytes
1741  * kmflag       Flag passed to kmem_alloc(9F); ignored in userland.
1742  */
1743 /* ARGSUSED */
1744 void *
1745 aes_alloc_keysched(size_t *size, int kmflag)
1746 {
1756 /* EXPORT DELETE START */
1747         aes_key_t *keysched;

1749 #ifdef  _KERNEL
1750         keysched = (aes_key_t *)kmem_alloc(sizeof (aes_key_t), kmflag);
1751 #else   /* !_KERNEL */
1752         keysched = (aes_key_t *)malloc(sizeof (aes_key_t));
1753 #endif  /* _KERNEL */

1755         if (keysched != NULL) {
1756                 *size = sizeof (aes_key_t);
1757                 return (keysched);
1758         }
1769 /* EXPORT DELETE END */
1759         return (NULL);
1760 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   27309 Thu Jul 11 01:28:59 2013
new/usr/src/common/crypto/aes/amd64/aes_amd64.s
first pass
**********************************************************
```

```
/*
 * ---------------------------------------------------------------------------
 * Copyright (c) 1998-2007, Brian Gladman, Worcester, UK. All rights reserved.
 *
 * LICENSE TERMS
 *
 * The free distribution and use of this software is allowed (with or without
 * changes) provided that:
 *
 *  1. source code distributions include the above copyright notice, this
 *     list of conditions and the following disclaimer;
 *
 *  2. binary distributions include the above copyright notice, this list
 *     of conditions and the following disclaimer in their documentation;
 *
 *  3. the name of the copyright holder is not used to endorse products
 *     built using this software without specific written permission.
 *
 * DISCLAIMER
 *
 * This software is provided 'as is' with no explicit or implied warranties
 * in respect of its properties, including, but not limited to, correctness
 * and/or fitness for purpose.
 * ---------------------------------------------------------------------------
 * Issue 20/12/2007
 *
 * I am grateful to Dag Arne Osvik for many discussions of the techniques that
 * can be used to optimise AES assembler code on AMD64/EM64T architectures.
 * Some of the techniques used in this implementation are the result of
 * suggestions made by him for which I am most grateful.
 *
 * An AES implementation for AMD64 processors using the YASM assembler.  This
 * implementation provides only encryption, decryption and hence requires key
 * scheduling support in C. It uses 8k bytes of tables but its encryption and
 * decryption performance is very close to that obtained using large tables.
 * It can use either MS Windows or Gnu/Linux/OpenSolaris OS calling conventions,
 * which are as follows:
 *                ms windows  gnu/linux/opensolaris os
 *
 *   in_blk          rcx     rdi
 *   out_blk         rdx     rsi
 *   context (cx)     r8     rdx
 *
 *   preserved       rsi      -    + rbx, rbp, rsp, r12, r13, r14 & r15
 *   registers       rdi      -      on both
 *
 *   destroyed        -      rsi   + rax, rcx, rdx, r8, r9, r10 & r11
 *   registers        -      rdi     on both
 *
 * The convention used here is that for gnu/linux/opensolaris os.
 *
 * This code provides the standard AES block size (128 bits, 16 bytes) and the
 * three standard AES key sizes (128, 192 and 256 bits). It has the same call
 * interface as my C implementation.  It uses the Microsoft C AMD64 calling
 * conventions in which the three parameters are placed in  rcx, rdx and r8
 * respectively.  The rbx, rsi, rdi, rbp and r12..r15 registers are preserved.
 *
 * OpenSolaris Note:
 * Modified to use GNU/Linux/Solaris calling conventions.
 * That is parameters are placed in rdi, rsi, rdx, and rcx, respectively.
 *
```



```
 *     AES_RETURN aes_encrypt(const unsigned char in_blk[],
 *                    unsigned char out_blk[], const aes_encrypt_ctx cx[1])/
 *
 *     AES_RETURN aes_decrypt(const unsigned char in_blk[],
 *                    unsigned char out_blk[], const aes_decrypt_ctx cx[1])/
 *
 *     AES_RETURN aes_encrypt_key<NNN>(const unsigned char key[],
 *                                           const aes_encrypt_ctx cx[1])/
 *
 *     AES_RETURN aes_decrypt_key<NNN>(const unsigned char key[],
 *                                           const aes_decrypt_ctx cx[1])/
 *
 *     AES_RETURN aes_encrypt_key(const unsigned char key[],
 *                              unsigned int len, const aes_decrypt_ctx cx[1])/
 *
 *     AES_RETURN aes_decrypt_key(const unsigned char key[],
 *                              unsigned int len, const aes_decrypt_ctx cx[1])/
 *
 * where <NNN> is 128, 102 or 256.  In the last two calls the length can be in
 * either bits or bytes.
 *
 * Comment in/out the following lines to obtain the desired subroutines. These
 * selections MUST match those in the C header file aesopt.h
 */
#define AES_REV_DKS       /* define if key decryption schedule is reversed */

#define LAST_ROUND_TABLES /* define for the faster version using extra tables */

/*
 * The encryption key schedule has the following in memory layout where N is the
 * number of rounds (10, 12 or 14):
 *
 * lo: | input key (round 0)  |  / each round is four 32-bit words
 *     | encryption round 1   |
 *     | encryption round 2   |
 *     ....
 *     | encryption round N-1 |
 * hi: | encryption round N   |
 *
 * The decryption key schedule is normally set up so that it has the same
 * layout as above by actually reversing the order of the encryption key
 * schedule in memory (this happens when AES_REV_DKS is set):
 *
 * lo: | decryption round 0   | =             | encryption round N   |
 *     | decryption round 1   | = INV_MIX_COL[ | encryption round N-1 | ]
 *     | decryption round 2   | = INV_MIX_COL[ | encryption round N-2 | ]
 *     ....                       ....
 *     | decryption round N-1 | = INV_MIX_COL[ | encryption round 1   | ]
 * hi: | decryption round N   | =             | input key (round 0)  |
 *
 * with rounds except the first and last modified using inv_mix_column()
 * But if AES_REV_DKS is NOT set the order of keys is left as it is for
 * encryption so that it has to be accessed in reverse when used for
 * decryption (although the inverse mix column modifications are done)
 *
 * lo: | decryption round 0   | =             | input key (round 0)  |
 *     | decryption round 1   | = INV_MIX_COL[ | encryption round 1   | ]
 *     | decryption round 2   | = INV_MIX_COL[ | encryption round 2   | ]
 *     ....                       ....
 *     | decryption round N-1 | = INV_MIX_COL[ | encryption round N-1 | ]
 * hi: | decryption round N   | =             | encryption round N   |
 *
 * This layout is faster when the assembler key scheduling provided here
 * is used.
 *
 * End of user defines
```

```
 128  */

 130 /*
 131  * ---------------------------------------------------------------------------
 132  * OpenSolaris OS modifications
 133  *
 134  * This source originates from Brian Gladman file aes_amd64.asm
 135  * in http://fp.gladman.plus.com/AES/aes-src-04-03-08.zip
 136  * with these changes:
 137  *
 138  * 1. Removed MS Windows-specific code within DLL_EXPORT, _SEH_, and
 139  * !__GNUC__ ifdefs.  Also removed ENCRYPTION, DECRYPTION,
 140  * AES_128, AES_192, AES_256, AES_VAR ifdefs.
 141  *
 142  * 2. Translate yasm/nasm %define and .macro definitions to cpp(1) #define
 143  *
 144  * 3. Translate yasm/nasm %ifdef/%ifndef to cpp(1) #ifdef
 145  *
 146  * 4. Translate Intel/yasm/nasm syntax to ATT/OpenSolaris as(1) syntax
 147  * (operands reversed, literals prefixed with "$", registers prefixed with "%",
 148  * and "[register+offset]", addressing changed to "offset(register)",
 149  * parenthesis in constant expressions "()" changed to square brackets "[]",
 150  * "." removed from  local (numeric) labels, and other changes.
 151  * Examples:
 152  * Intel/yasm/nasm Syntax       ATT/OpenSolaris Syntax
 153  * mov  rax,(4*20h)             mov     $[4*0x20],%rax
 154  * mov  rax,[ebx+20h]           mov     0x20(%ebx),%rax
 155  * lea  rax,[ebx+ecx]           lea     (%ebx,%ecx),%rax
 156  * sub  rax,[ebx+ecx*4-20h]     sub     -0x20(%ebx,%ecx,4),%rax
 157  *
 158  * 5. Added OpenSolaris ENTRY_NP/SET_SIZE macros from
 159  * /usr/include/sys/asm_linkage.h, lint(1B) guards, and dummy C function
 160  * definitions for lint.
 159  * /usr/include/sys/asm_linkage.h, lint(1B) guards, EXPORT DELETE START
 160  * and EXPORT DELETE END markers, and dummy C function definitions for lint.
 161  *
 162  * 6. Renamed functions and reordered parameters to match OpenSolaris:
 163  * Original Gladman interface:
 164  *      int aes_encrypt(const unsigned char *in,
 165  *              unsigned char *out, const aes_encrypt_ctx cx[1])/
 166  *      int aes_decrypt(const unsigned char *in,
 167  *              unsigned char *out, const aes_encrypt_ctx cx[1])/
 168  * Note: aes_encrypt_ctx contains ks, a 60 element array of uint32_t,
 169  * and a union type, inf., containing inf.l, a uint32_t and
 170  * inf.b, a 4-element array of uint32_t.  Only b[0] in the array (aka "l") is
 171  * used and contains the key schedule length * 16 where key schedule length is
 172  * 10, 12, or 14 bytes.
 173  *
 174  * OpenSolaris OS interface:
 175  *      void aes_encrypt_amd64(const aes_ks_t *ks, int Nr,
 176  *              const uint32_t pt[4], uint32_t ct[4])/
 177  *      void aes_decrypt_amd64(const aes_ks_t *ks, int Nr,
 178  *              const uint32_t pt[4], uint32_t ct[4])/
 179  *      typedef union {uint64_t ks64[(MAX_AES_NR + 1) * 4]/
 180  *               uint32_t ks32[(MAX_AES_NR + 1) * 4]/ } aes_ks_t/
 181  * Note: ks is the AES key schedule, Nr is number of rounds, pt is plain text,
 182  * ct is crypto text, and MAX_AES_NR is 14.
 183  * For the x86 64-bit architecture, OpenSolaris OS uses ks32 instead of ks64.
 184  */

 186 #if defined(lint) || defined(__lint)

 188 #include <sys/types.h>
 189 /* ARGSUSED */
 190 void
 191 aes_encrypt_amd64(const uint32_t rk[], int Nr, const uint32_t pt[4],
```



```
 192         uint32_t ct[4]) {
 193 }
_______unchanged_portion_omitted_


 201 #else

 203 #include <sys/asm_linkage.h>

 205 #define KS_LENGTH       60

 207 #define raxd            eax
 208 #define rdxd            edx
 209 #define rcxd            ecx
 210 #define rbxd            ebx
 211 #define rsid            esi
 212 #define rdid            edi

 214 #define raxb            al
 215 #define rdxb            dl
 216 #define rcxb            cl
 217 #define rbxb            bl
 218 #define rsib            sil
 219 #define rdib            dil

 221 / finite field multiplies by {02}, {04} and {08}

 223 #define f2(x) [[x<<1]^[[[x>>7]&1]*0x11b]]
 224 #define f4(x) [[x<<2]^[[[x>>6]&1]*0x11b]^[[[x>>6]&2]*0x11b]]
 225 #define f8(x) [[x<<3]^[[[x>>5]&1]*0x11b]^[[[x>>5]&2]*0x11b]^[[[x>>5]&4]*0x11b]]

 227 / finite field multiplies required in table generation

 229 #define f3(x) [[f2(x)] ^ [x]]
 230 #define f9(x) [[f8(x)] ^ [x]]
 231 #define fb(x) [[f8(x)] ^ [f2(x)] ^ [x]]
 232 #define fd(x) [[f8(x)] ^ [f4(x)] ^ [x]]
 233 #define fe(x) [[f8(x)] ^ [f4(x)] ^ [f2(x)]]

 235 / macros for expanding S-box data

 237 #define u8(x) [f2(x)], [x], [x], [f3(x)], [f2(x)], [x], [x], [f3(x)]
 238 #define v8(x) [fe(x)], [f9(x)], [fd(x)], [fb(x)], [fe(x)], [f9(x)], [fd(x)], [x]
 239 #define w8(x) [x], 0, 0, 0, [x], 0, 0, 0

 241 #define enc_vals(x)     \
 242    .byte x(0x63),x(0x7c),x(0x77),x(0x7b),x(0xf2),x(0x6b),x(0x6f),x(0xc5); \
 243    .byte x(0x30),x(0x01),x(0x67),x(0x2b),x(0xfe),x(0xd7),x(0xab),x(0x76); \
 244    .byte x(0xca),x(0x82),x(0xc9),x(0x7d),x(0xfa),x(0x59),x(0x47),x(0xf0); \
 245    .byte x(0xad),x(0xd4),x(0xa2),x(0xaf),x(0x9c),x(0xa4),x(0x72),x(0xc0); \
 246    .byte x(0xb7),x(0xfd),x(0x93),x(0x26),x(0x36),x(0x3f),x(0xf7),x(0xcc); \
 247    .byte x(0x34),x(0xa5),x(0xe5),x(0xf1),x(0x71),x(0xd8),x(0x31),x(0x15); \
 248    .byte x(0x04),x(0xc7),x(0x23),x(0xc3),x(0x18),x(0x96),x(0x05),x(0x9a); \
 249    .byte x(0x07),x(0x12),x(0x80),x(0xe2),x(0xeb),x(0x27),x(0xb2),x(0x75); \
 250    .byte x(0x09),x(0x83),x(0x2c),x(0x1a),x(0x1b),x(0x6e),x(0x5a),x(0xa0); \
 251    .byte x(0x52),x(0x3b),x(0xd6),x(0xb3),x(0x29),x(0xe3),x(0x2f),x(0x84); \
 252    .byte x(0x53),x(0xd1),x(0x00),x(0xed),x(0x20),x(0xfc),x(0xb1),x(0x5b); \
 253    .byte x(0x6a),x(0xcb),x(0xbe),x(0x39),x(0x4a),x(0x4c),x(0x58),x(0xcf); \
 254    .byte x(0xd0),x(0xef),x(0xaa),x(0xfb),x(0x43),x(0x4d),x(0x33),x(0x85); \
 255    .byte x(0x45),x(0xf9),x(0x02),x(0x7f),x(0x50),x(0x3c),x(0x9f),x(0xa8); \
 256    .byte x(0x51),x(0xa3),x(0x40),x(0x8f),x(0x92),x(0x9d),x(0x38),x(0xf5); \
 257    .byte x(0xbc),x(0xb6),x(0xda),x(0x21),x(0x10),x(0xff),x(0xf3),x(0xd2); \
 258    .byte x(0xcd),x(0x0c),x(0x13),x(0xec),x(0x5f),x(0x97),x(0x44),x(0x17); \
 259    .byte x(0xc4),x(0xa7),x(0x7e),x(0x3d),x(0x64),x(0x5d),x(0x19),x(0x73); \
 260    .byte x(0x60),x(0x81),x(0x4f),x(0xdc),x(0x22),x(0x2a),x(0x90),x(0x88); \
 261    .byte x(0x46),x(0xee),x(0xb8),x(0x14),x(0xde),x(0x5e),x(0x0b),x(0xdb); \
```

```
 262    .byte x(0xe0),x(0x32),x(0x3a),x(0x0a),x(0x49),x(0x06),x(0x24),x(0x5c); \
 263    .byte x(0xc2),x(0xd3),x(0xac),x(0x62),x(0x91),x(0x95),x(0xe4),x(0x79); \
 264    .byte x(0xe7),x(0xc8),x(0x37),x(0x6d),x(0x8d),x(0xd5),x(0x4e),x(0xa9); \
 265    .byte x(0x6c),x(0x56),x(0xf4),x(0xea),x(0x65),x(0x7a),x(0xae),x(0x08); \
 266    .byte x(0xba),x(0x78),x(0x25),x(0x2e),x(0x1c),x(0xa6),x(0xb4),x(0xc6); \
 267    .byte x(0xe8),x(0xdd),x(0x74),x(0x1f),x(0x4b),x(0xbd),x(0x8b),x(0x8a); \
 268    .byte x(0x70),x(0x3e),x(0xb5),x(0x66),x(0x48),x(0x03),x(0xf6),x(0x0e); \
 269    .byte x(0x61),x(0x35),x(0x57),x(0xb9),x(0x86),x(0xc1),x(0x1d),x(0x9e); \
 270    .byte x(0xe1),x(0xf8),x(0x98),x(0x11),x(0x69),x(0xd9),x(0x8e),x(0x94); \
 271    .byte x(0x9b),x(0x1e),x(0x87),x(0xe9),x(0xce),x(0x55),x(0x28),x(0xdf); \
 272    .byte x(0x8c),x(0xa1),x(0x89),x(0x0d),x(0xbf),x(0xe6),x(0x42),x(0x68); \
 273    .byte x(0x41),x(0x99),x(0x2d),x(0x0f),x(0xb0),x(0x54),x(0xbb),x(0x16)

 275 #define dec_vals(x) \
 276    .byte x(0x52),x(0x09),x(0x6a),x(0xd5),x(0x30),x(0x36),x(0xa5),x(0x38); \
 277    .byte x(0xbf),x(0x40),x(0xa3),x(0x9e),x(0x81),x(0xf3),x(0xd7),x(0xfb); \
 278    .byte x(0x7c),x(0xe3),x(0x39),x(0x82),x(0x9b),x(0x2f),x(0xff),x(0x87); \
 279    .byte x(0x34),x(0x8e),x(0x43),x(0x44),x(0xc4),x(0xde),x(0xe9),x(0xcb); \
 280    .byte x(0x54),x(0x7b),x(0x94),x(0x32),x(0xa6),x(0xc2),x(0x23),x(0x3d); \
 281    .byte x(0xee),x(0x4c),x(0x95),x(0x0b),x(0x42),x(0xfa),x(0xc3),x(0x4e); \
 282    .byte x(0x08),x(0x2e),x(0xa1),x(0x66),x(0x28),x(0xd9),x(0x24),x(0xb2); \
 283    .byte x(0x76),x(0x5b),x(0xa2),x(0x49),x(0x6d),x(0x8b),x(0xd1),x(0x25); \
 284    .byte x(0x72),x(0xf8),x(0xf6),x(0x64),x(0x86),x(0x68),x(0x98),x(0x16); \
 285    .byte x(0xd4),x(0xa4),x(0x5c),x(0xcc),x(0x5d),x(0x65),x(0xb6),x(0x92); \
 286    .byte x(0x6c),x(0x70),x(0x48),x(0x50),x(0xfd),x(0xed),x(0xb9),x(0xda); \
 287    .byte x(0x5e),x(0x15),x(0x46),x(0x57),x(0xa7),x(0x8d),x(0x9d),x(0x84); \
 288    .byte x(0x90),x(0xd8),x(0xab),x(0x00),x(0x8c),x(0xbc),x(0xd3),x(0x0a); \
 289    .byte x(0xf7),x(0xe4),x(0x58),x(0x05),x(0xb8),x(0xb3),x(0x45),x(0x06); \
 290    .byte x(0xd0),x(0x2c),x(0x1e),x(0x8f),x(0xca),x(0x3f),x(0x0f),x(0x02); \
 291    .byte x(0xc1),x(0xaf),x(0xbd),x(0x03),x(0x01),x(0x13),x(0x8a),x(0x6b); \
 292    .byte x(0x3a),x(0x91),x(0x11),x(0x41),x(0x4f),x(0x67),x(0xdc),x(0xea); \
 293    .byte x(0x97),x(0xf2),x(0xcf),x(0xce),x(0xf0),x(0xb4),x(0xe6),x(0x73); \
 294    .byte x(0x96),x(0xac),x(0x74),x(0x22),x(0xe7),x(0xad),x(0x35),x(0x85); \
 295    .byte x(0xe2),x(0xf9),x(0x37),x(0xe8),x(0x1c),x(0x75),x(0xdf),x(0x6e); \
 296    .byte x(0x47),x(0xf1),x(0x1a),x(0x71),x(0x1d),x(0x29),x(0xc5),x(0x89); \
 297    .byte x(0x6f),x(0xb7),x(0x62),x(0x0e),x(0xaa),x(0x18),x(0xbe),x(0x1b); \
 298    .byte x(0xfc),x(0x56),x(0x3e),x(0x4b),x(0xc6),x(0xd2),x(0x79),x(0x20); \
 299    .byte x(0x9a),x(0xdb),x(0xc0),x(0xfe),x(0x78),x(0xcd),x(0x5a),x(0xf4); \
 300    .byte x(0x1f),x(0xdd),x(0xa8),x(0x33),x(0x88),x(0x07),x(0xc7),x(0x31); \
 301    .byte x(0xb1),x(0x12),x(0x10),x(0x59),x(0x27),x(0x80),x(0xec),x(0x5f); \
 302    .byte x(0x60),x(0x51),x(0x7f),x(0xa9),x(0x19),x(0xb5),x(0x4a),x(0x0d); \
 303    .byte x(0x2d),x(0xe5),x(0x7a),x(0x9f),x(0x93),x(0xc9),x(0x9c),x(0xef); \
 304    .byte x(0xa0),x(0xe0),x(0x3b),x(0x4d),x(0xae),x(0x2a),x(0xf5),x(0xb0); \
 305    .byte x(0xc8),x(0xeb),x(0xbb),x(0x3c),x(0x83),x(0x53),x(0x99),x(0x61); \
 306    .byte x(0x17),x(0x2b),x(0x04),x(0x7e),x(0xba),x(0x77),x(0xd6),x(0x26); \
 307    .byte x(0xe1),x(0x69),x(0x14),x(0x63),x(0x55),x(0x21),x(0x0c),x(0x7d)

 309 #define tptr    %rbp    /* table pointer */
 310 #define kptr    %r8     /* key schedule pointer */
 311 #define fofs    128     /* adjust offset in key schedule to keep |disp| < 128 */
 312 #define fk_ref(x, y)    -16*x+fofs+4*y(kptr)

 314 #ifdef  AES_REV_DKS
 315 #define rofs            128
 316 #define ik_ref(x, y)    -16*x+rofs+4*y(kptr)

 318 #else
 319 #define rofs            -128
 320 #define ik_ref(x, y)    16*x+rofs+4*y(kptr)
 321 #endif  /* AES_REV_DKS */

 323 #define tab_0(x)        (tptr,x,8)
 324 #define tab_1(x)        3(tptr,x,8)
 325 #define tab_2(x)        2(tptr,x,8)
 326 #define tab_3(x)        1(tptr,x,8)
 327 #define tab_f(x)        1(tptr,x,8)
```



```
 328 #define tab_i(x)        7(tptr,x,8)

 330         /* EXPORT DELETE START */
 330 #define ff_rnd(p1, p2, p3, p4, round)   /* normal forward round */ \
 331         mov     fk_ref(round,0), p1; \
 332         mov     fk_ref(round,1), p2; \
 333         mov     fk_ref(round,2), p3; \
 334         mov     fk_ref(round,3), p4; \
 335  \
 336         movzx   %al, %esi; \
 337         movzx   %ah, %edi; \
 338         shr     $16, %eax; \
 339         xor     tab_0(%rsi), p1; \
 340         xor     tab_1(%rdi), p4; \
 341         movzx   %al, %esi; \
 342         movzx   %ah, %edi; \
 343         xor     tab_2(%rsi), p3; \
 344         xor     tab_3(%rdi), p2; \
 345  \
 346         movzx   %bl, %esi; \
 347         movzx   %bh, %edi; \
 348         shr     $16, %ebx; \
 349         xor     tab_0(%rsi), p2; \
 350         xor     tab_1(%rdi), p1; \
 351         movzx   %bl, %esi; \
 352         movzx   %bh, %edi; \
 353         xor     tab_2(%rsi), p4; \
 354         xor     tab_3(%rdi), p3; \
 355  \
 356         movzx   %cl, %esi; \
 357         movzx   %ch, %edi; \
 358         shr     $16, %ecx; \
 359         xor     tab_0(%rsi), p3; \
 360         xor     tab_1(%rdi), p2; \
 361         movzx   %cl, %esi; \
 362         movzx   %ch, %edi; \
 363         xor     tab_2(%rsi), p1; \
 364         xor     tab_3(%rdi), p4; \
 365  \
 366         movzx   %dl, %esi; \
 367         movzx   %dh, %edi; \
 368         shr     $16, %edx; \
 369         xor     tab_0(%rsi), p4; \
 370         xor     tab_1(%rdi), p3; \
 371         movzx   %dl, %esi; \
 372         movzx   %dh, %edi; \
 373         xor     tab_2(%rsi), p2; \
 374         xor     tab_3(%rdi), p1; \
 375  \
 376         mov     p1, %eax; \
 377         mov     p2, %ebx; \
 378         mov     p3, %ecx; \
 379         mov     p4, %edx

 381 #ifdef  LAST_ROUND_TABLES

 383 #define fl_rnd(p1, p2, p3, p4, round)   /* last forward round */ \
 384         add     $2048, tptr; \
 385         mov     fk_ref(round,0), p1; \
 386         mov     fk_ref(round,1), p2; \
 387         mov     fk_ref(round,2), p3; \
 388         mov     fk_ref(round,3), p4; \
 389  \
 390         movzx   %al, %esi; \
 391         movzx   %ah, %edi; \
 392         shr     $16, %eax; \
```

```
393         xor     tab_0(%rsi), p1; \
394         xor     tab_1(%rdi), p4; \
395         movzx   %al, %esi; \
396         movzx   %ah, %edi; \
397         xor     tab_2(%rsi), p3; \
398         xor     tab_3(%rdi), p2; \
399  \
400         movzx   %bl, %esi; \
401         movzx   %bh, %edi; \
402         shr     $16, %ebx; \
403         xor     tab_0(%rsi), p2; \
404         xor     tab_1(%rdi), p1; \
405         movzx   %bl, %esi; \
406         movzx   %bh, %edi; \
407         xor     tab_2(%rsi), p4; \
408         xor     tab_3(%rdi), p3; \
409  \
410         movzx   %cl, %esi; \
411         movzx   %ch, %edi; \
412         shr     $16, %ecx; \
413         xor     tab_0(%rsi), p3; \
414         xor     tab_1(%rdi), p2; \
415         movzx   %cl, %esi; \
416         movzx   %ch, %edi; \
417         xor     tab_2(%rsi), p1; \
418         xor     tab_3(%rdi), p4; \
419  \
420         movzx   %dl, %esi; \
421         movzx   %dh, %edi; \
422         shr     $16, %edx; \
423         xor     tab_0(%rsi), p4; \
424         xor     tab_1(%rdi), p3; \
425         movzx   %dl, %esi; \
426         movzx   %dh, %edi; \
427         xor     tab_2(%rsi), p2; \
428         xor     tab_3(%rdi), p1

430 #else

432 #define fl_rnd(p1, p2, p3, p4, round)   /* last forward round */ \
433         mov     fk_ref(round,0), p1; \
434         mov     fk_ref(round,1), p2; \
435         mov     fk_ref(round,2), p3; \
436         mov     fk_ref(round,3), p4; \
437  \
438         movzx   %al, %esi; \
439         movzx   %ah, %edi; \
440         shr     $16, %eax; \
441         movzx   tab_f(%rsi), %esi; \
442         movzx   tab_f(%rdi), %edi; \
443         xor     %esi, p1; \
444         rol     $8, %edi; \
445         xor     %edi, p4; \
446         movzx   %al, %esi; \
447         movzx   %ah, %edi; \
448         movzx   tab_f(%rsi), %esi; \
449         movzx   tab_f(%rdi), %edi; \
450         rol     $16, %esi; \
451         rol     $24, %edi; \
452         xor     %esi, p3; \
453         xor     %edi, p2; \
454  \
455         movzx   %bl, %esi; \
456         movzx   %bh, %edi; \
457         shr     $16, %ebx; \
458         movzx   tab_f(%rsi), %esi; \
```



```
459         movzx   tab_f(%rdi), %edi; \
460         xor     %esi, p2; \
461         rol     $8, %edi; \
462         xor     %edi, p1; \
463         movzx   %bl, %esi; \
464         movzx   %bh, %edi; \
465         movzx   tab_f(%rsi), %esi; \
466         movzx   tab_f(%rdi), %edi; \
467         rol     $16, %esi; \
468         rol     $24, %edi; \
469         xor     %esi, p4; \
470         xor     %edi, p3; \
471  \
472         movzx   %cl, %esi; \
473         movzx   %ch, %edi; \
474         movzx   tab_f(%rsi), %esi; \
475         movzx   tab_f(%rdi), %edi; \
476         shr     $16, %ecx; \
477         xor     %esi, p3; \
478         rol     $8, %edi; \
479         xor     %edi, p2; \
480         movzx   %cl, %esi; \
481         movzx   %ch, %edi; \
482         movzx   tab_f(%rsi), %esi; \
483         movzx   tab_f(%rdi), %edi; \
484         rol     $16, %esi; \
485         rol     $24, %edi; \
486         xor     %esi, p1; \
487         xor     %edi, p4; \
488  \
489         movzx   %dl, %esi; \
490         movzx   %dh, %edi; \
491         movzx   tab_f(%rsi), %esi; \
492         movzx   tab_f(%rdi), %edi; \
493         shr     $16, %edx; \
494         xor     %esi, p4; \
495         rol     $8, %edi; \
496         xor     %edi, p3; \
497         movzx   %dl, %esi; \
498         movzx   %dh, %edi; \
499         movzx   tab_f(%rsi), %esi; \
500         movzx   tab_f(%rdi), %edi; \
501         rol     $16, %esi; \
502         rol     $24, %edi; \
503         xor     %esi, p2; \
504         xor     %edi, p1

506 #endif  /* LAST_ROUND_TABLES */

508 #define ii_rnd(p1, p2, p3, p4, round)   /* normal inverse round */ \
509         mov     ik_ref(round,0), p1; \
510         mov     ik_ref(round,1), p2; \
511         mov     ik_ref(round,2), p3; \
512         mov     ik_ref(round,3), p4; \
513  \
514         movzx   %al, %esi; \
515         movzx   %ah, %edi; \
516         shr     $16, %eax; \
517         xor     tab_0(%rsi), p1; \
518         xor     tab_1(%rdi), p2; \
519         movzx   %al, %esi; \
520         movzx   %ah, %edi; \
521         xor     tab_2(%rsi), p3; \
522         xor     tab_3(%rdi), p4; \
523  \
524         movzx   %bl, %esi; \
```

```
525         movzx   %bh, %edi; \
526         shr     $16, %ebx; \
527         xor     tab_0(%rsi), p2; \
528         xor     tab_1(%rdi), p3; \
529         movzx   %bl, %esi; \
530         movzx   %bh, %edi; \
531         xor     tab_2(%rsi), p4; \
532         xor     tab_3(%rdi), p1; \
533  \
534         movzx   %cl, %esi; \
535         movzx   %ch, %edi; \
536         shr     $16, %ecx; \
537         xor     tab_0(%rsi), p3; \
538         xor     tab_1(%rdi), p4; \
539         movzx   %cl, %esi; \
540         movzx   %ch, %edi; \
541         xor     tab_2(%rsi), p1; \
542         xor     tab_3(%rdi), p2; \
543  \
544         movzx   %dl, %esi; \
545         movzx   %dh, %edi; \
546         shr     $16, %edx; \
547         xor     tab_0(%rsi), p4; \
548         xor     tab_1(%rdi), p1; \
549         movzx   %dl, %esi; \
550         movzx   %dh, %edi; \
551         xor     tab_2(%rsi), p2; \
552         xor     tab_3(%rdi), p3; \
553  \
554         mov     p1, %eax; \
555         mov     p2, %ebx; \
556         mov     p3, %ecx; \
557         mov     p4, %edx

559 #ifdef  LAST_ROUND_TABLES

561 #define il_rnd(p1, p2, p3, p4, round)   /* last inverse round */ \
562         add     $2048, tptr; \
563         mov     ik_ref(round,0), p1; \
564         mov     ik_ref(round,1), p2; \
565         mov     ik_ref(round,2), p3; \
566         mov     ik_ref(round,3), p4; \
567  \
568         movzx   %al, %esi; \
569         movzx   %ah, %edi; \
570         shr     $16, %eax; \
571         xor     tab_0(%rsi), p1; \
572         xor     tab_1(%rdi), p2; \
573         movzx   %al, %esi; \
574         movzx   %ah, %edi; \
575         xor     tab_2(%rsi), p3; \
576         xor     tab_3(%rdi), p4; \
577  \
578         movzx   %bl, %esi; \
579         movzx   %bh, %edi; \
580         shr     $16, %ebx; \
581         xor     tab_0(%rsi), p2; \
582         xor     tab_1(%rdi), p3; \
583         movzx   %bl, %esi; \
584         movzx   %bh, %edi; \
585         xor     tab_2(%rsi), p4; \
586         xor     tab_3(%rdi), p1; \
587  \
588         movzx   %cl, %esi; \
589         movzx   %ch, %edi; \
590         shr     $16, %ecx; \
```



```
591         xor     tab_0(%rsi), p3; \
592         xor     tab_1(%rdi), p4; \
593         movzx   %cl, %esi; \
594         movzx   %ch, %edi; \
595         xor     tab_2(%rsi), p1; \
596         xor     tab_3(%rdi), p2; \
597  \
598         movzx   %dl, %esi; \
599         movzx   %dh, %edi; \
600         shr     $16, %edx; \
601         xor     tab_0(%rsi), p4; \
602         xor     tab_1(%rdi), p1; \
603         movzx   %dl, %esi; \
604         movzx   %dh, %edi; \
605         xor     tab_2(%rsi), p2; \
606         xor     tab_3(%rdi), p3

608 #else

610 #define il_rnd(p1, p2, p3, p4, round)   /* last inverse round */ \
611         mov     ik_ref(round,0), p1; \
612         mov     ik_ref(round,1), p2; \
613         mov     ik_ref(round,2), p3; \
614         mov     ik_ref(round,3), p4; \
615  \
616         movzx   %al, %esi; \
617         movzx   %ah, %edi; \
618         movzx   tab_i(%rsi), %esi; \
619         movzx   tab_i(%rdi), %edi; \
620         shr     $16, %eax; \
621         xor     %esi, p1; \
622         rol     $8, %edi; \
623         xor     %edi, p2; \
624         movzx   %al, %esi; \
625         movzx   %ah, %edi; \
626         movzx   tab_i(%rsi), %esi; \
627         movzx   tab_i(%rdi), %edi; \
628         rol     $16, %esi; \
629         rol     $24, %edi; \
630         xor     %esi, p3; \
631         xor     %edi, p4; \
632  \
633         movzx   %bl, %esi; \
634         movzx   %bh, %edi; \
635         movzx   tab_i(%rsi), %esi; \
636         movzx   tab_i(%rdi), %edi; \
637         shr     $16, %ebx; \
638         xor     %esi, p2; \
639         rol     $8, %edi; \
640         xor     %edi, p3; \
641         movzx   %bl, %esi; \
642         movzx   %bh, %edi; \
643         movzx   tab_i(%rsi), %esi; \
644         movzx   tab_i(%rdi), %edi; \
645         rol     $16, %esi; \
646         rol     $24, %edi; \
647         xor     %esi, p4; \
648         xor     %edi, p1; \
649  \
650         movzx   %cl, %esi; \
651         movzx   %ch, %edi; \
652         movzx   tab_i(%rsi), %esi; \
653         movzx   tab_i(%rdi), %edi; \
654         shr     $16, %ecx; \
655         xor     %esi, p3; \
656         rol     $8, %edi; \
```

```
657         xor     %edi, p4; \
658         movzx   %cl, %esi; \
659         movzx   %ch, %edi; \
660         movzx   tab_i(%rsi), %esi; \
661         movzx   tab_i(%rdi), %edi; \
662         rol     $16, %esi; \
663         rol     $24, %edi; \
664         xor     %esi, p1; \
665         xor     %edi, p2; \
666  \
667         movzx   %dl, %esi; \
668         movzx   %dh, %edi; \
669         movzx   tab_i(%rsi), %esi; \
670         movzx   tab_i(%rdi), %edi; \
671         shr     $16, %edx; \
672         xor     %esi, p4; \
673         rol     $8, %edi; \
674         xor     %edi, p1; \
675         movzx   %dl, %esi; \
676         movzx   %dh, %edi; \
677         movzx   tab_i(%rsi), %esi; \
678         movzx   tab_i(%rdi), %edi; \
679         rol     $16, %esi; \
680         rol     $24, %edi; \
681         xor     %esi, p2; \
682         xor     %edi, p3

684 #endif  /* LAST_ROUND_TABLES */
686         /* EXPORT DELETE END */

686 /*
687  * OpenSolaris OS:
688  * void aes_encrypt_amd64(const aes_ks_t *ks, int Nr,
689  *      const uint32_t pt[4], uint32_t ct[4])/
690  *
691  * Original interface:
692  * int aes_encrypt(const unsigned char *in,
693  *      unsigned char *out, const aes_encrypt_ctx cx[1])/
694  */
695         .align  64
696 enc_tab:
697         enc_vals(u8)
698 #ifdef  LAST_ROUND_TABLES
699         / Last Round Tables:
700         enc_vals(w8)
701 #endif


704         ENTRY_NP(aes_encrypt_amd64)
707         /* EXPORT DELETE START */
705 #ifdef  GLADMAN_INTERFACE
706         / Original interface
707         sub     $[4*8], %rsp    / gnu/linux/opensolaris binary interface
708         mov     %rsi, (%rsp)    / output pointer (P2)
709         mov     %rdx, %r8       / context (P3)

711         mov     %rbx, 1*8(%rsp) / P1: input pointer in rdi
712         mov     %rbp, 2*8(%rsp) / P2: output pointer in (rsp)
713         mov     %r12, 3*8(%rsp) / P3: context in r8
714         movzx   4*KS_LENGTH(kptr), %esi / Get byte key length * 16

716 #else
717         / OpenSolaris OS interface
718         sub     $[4*8], %rsp    / Make room on stack to save registers
719         mov     %rcx, (%rsp)    / Save output pointer (P4) on stack
720         mov     %rdi, %r8       / context (P1)
```



```
721         mov     %rdx, %rdi      / P3: save input pointer
722         shl     $4, %esi        / P2: esi byte key length * 16

724         mov     %rbx, 1*8(%rsp) / Save registers
725         mov     %rbp, 2*8(%rsp)
726         mov     %r12, 3*8(%rsp)
727         / P1: context in r8
728         / P2: byte key length * 16 in esi
729         / P3: input pointer in rdi
730         / P4: output pointer in (rsp)
731 #endif  /* GLADMAN_INTERFACE */

733         lea     enc_tab(%rip), tptr
734         sub     $fofs, kptr

736         / Load input block into registers
737         mov     (%rdi), %eax
738         mov     1*4(%rdi), %ebx
739         mov     2*4(%rdi), %ecx
740         mov     3*4(%rdi), %edx

742         xor     fofs(kptr), %eax
743         xor     fofs+4(kptr), %ebx
744         xor     fofs+8(kptr), %ecx
745         xor     fofs+12(kptr), %edx

747         lea     (kptr,%rsi), kptr
748         / Jump based on byte key length * 16:
749         cmp     $[10*16], %esi
750         je      3f
751         cmp     $[12*16], %esi
752         je      2f
753         cmp     $[14*16], %esi
754         je      1f
755         mov     $-1, %rax       / error
756         jmp     4f

758         / Perform normal forward rounds
759 1:      ff_rnd(%r9d, %r10d, %r11d, %r12d, 13)
760         ff_rnd(%r9d, %r10d, %r11d, %r12d, 12)
761 2:      ff_rnd(%r9d, %r10d, %r11d, %r12d, 11)
762         ff_rnd(%r9d, %r10d, %r11d, %r12d, 10)
763 3:      ff_rnd(%r9d, %r10d, %r11d, %r12d,  9)
764         ff_rnd(%r9d, %r10d, %r11d, %r12d,  8)
765         ff_rnd(%r9d, %r10d, %r11d, %r12d,  7)
766         ff_rnd(%r9d, %r10d, %r11d, %r12d,  6)
767         ff_rnd(%r9d, %r10d, %r11d, %r12d,  5)
768         ff_rnd(%r9d, %r10d, %r11d, %r12d,  4)
769         ff_rnd(%r9d, %r10d, %r11d, %r12d,  3)
770         ff_rnd(%r9d, %r10d, %r11d, %r12d,  2)
771         ff_rnd(%r9d, %r10d, %r11d, %r12d,  1)
772         fl_rnd(%r9d, %r10d, %r11d, %r12d,  0)

774         / Copy results
775         mov     (%rsp), %rbx
776         mov     %r9d, (%rbx)
777         mov     %r10d, 4(%rbx)
778         mov     %r11d, 8(%rbx)
779         mov     %r12d, 12(%rbx)
780         xor     %rax, %rax
781 4:      / Restore registers
782         mov     1*8(%rsp), %rbx
783         mov     2*8(%rsp), %rbp
784         mov     3*8(%rsp), %r12
785         add     $[4*8], %rsp
789         /* EXPORT DELETE END */
```

```
786         ret

788         SET_SIZE(aes_encrypt_amd64)

790 /*
791  * OpenSolaris OS:
792  * void aes_decrypt_amd64(const aes_ks_t *ks, int Nr,
793  *      const uint32_t pt[4], uint32_t ct[4])/
794  *
795  * Original interface:
796  * int aes_decrypt(const unsigned char *in,
797  *      unsigned char *out, const aes_encrypt_ctx cx[1])/
798  */
799         .align  64
800 dec_tab:
801         dec_vals(v8)
802 #ifdef  LAST_ROUND_TABLES
803         / Last Round Tables:
804         dec_vals(w8)
805 #endif


808         ENTRY_NP(aes_decrypt_amd64)
813         /* EXPORT DELETE START */
809 #ifdef  GLADMAN_INTERFACE
810         / Original interface
811         sub     $[4*8], %rsp    / gnu/linux/opensolaris binary interface
812         mov     %rsi, (%rsp)    / output pointer (P2)
813         mov     %rdx, %r8       / context (P3)

815         mov     %rbx, 1*8(%rsp) / P1: input pointer in rdi
816         mov     %rbp, 2*8(%rsp) / P2: output pointer in (rsp)
817         mov     %r12, 3*8(%rsp) / P3: context in r8
818         movzx   4*KS_LENGTH(kptr), %esi / Get byte key length * 16

820 #else
821         / OpenSolaris OS interface
822         sub     $[4*8], %rsp    / Make room on stack to save registers
823         mov     %rcx, (%rsp)    / Save output pointer (P4) on stack
824         mov     %rdi, %r8       / context (P1)
825         mov     %rdx, %rdi      / P3: save input pointer
826         shl     $4, %esi        / P2: esi byte key length * 16

828         mov     %rbx, 1*8(%rsp) / Save registers
829         mov     %rbp, 2*8(%rsp)
830         mov     %r12, 3*8(%rsp)
831         / P1: context in r8
832         / P2: byte key length * 16 in esi
833         / P3: input pointer in rdi
834         / P4: output pointer in (rsp)
835 #endif  /* GLADMAN_INTERFACE */

837         lea     dec_tab(%rip), tptr
838         sub     $rofs, kptr

840         / Load input block into registers
841         mov     (%rdi), %eax
842         mov     1*4(%rdi), %ebx
843         mov     2*4(%rdi), %ecx
844         mov     3*4(%rdi), %edx

846 #ifdef AES_REV_DKS
847         mov     kptr, %rdi
848         lea     (kptr,%rsi), kptr
849 #else
850         lea     (kptr,%rsi), %rdi
```



```
851 #endif

853         xor     rofs(%rdi), %eax
854         xor     rofs+4(%rdi), %ebx
855         xor     rofs+8(%rdi), %ecx
856         xor     rofs+12(%rdi), %edx

858         / Jump based on byte key length * 16:
859         cmp     $[10*16], %esi
860         je      3f
861         cmp     $[12*16], %esi
862         je      2f
863         cmp     $[14*16], %esi
864         je      1f
865         mov     $-1, %rax       / error
866         jmp     4f

868         / Perform normal inverse rounds
869 1:      ii_rnd(%r9d, %r10d, %r11d, %r12d, 13)
870         ii_rnd(%r9d, %r10d, %r11d, %r12d, 12)
871 2:      ii_rnd(%r9d, %r10d, %r11d, %r12d, 11)
872         ii_rnd(%r9d, %r10d, %r11d, %r12d, 10)
873 3:      ii_rnd(%r9d, %r10d, %r11d, %r12d,  9)
874         ii_rnd(%r9d, %r10d, %r11d, %r12d,  8)
875         ii_rnd(%r9d, %r10d, %r11d, %r12d,  7)
876         ii_rnd(%r9d, %r10d, %r11d, %r12d,  6)
877         ii_rnd(%r9d, %r10d, %r11d, %r12d,  5)
878         ii_rnd(%r9d, %r10d, %r11d, %r12d,  4)
879         ii_rnd(%r9d, %r10d, %r11d, %r12d,  3)
880         ii_rnd(%r9d, %r10d, %r11d, %r12d,  2)
881         ii_rnd(%r9d, %r10d, %r11d, %r12d,  1)
882         il_rnd(%r9d, %r10d, %r11d, %r12d,  0)

884         / Copy results
885         mov     (%rsp), %rbx
886         mov     %r9d, (%rbx)
887         mov     %r10d, 4(%rbx)
888         mov     %r11d, 8(%rbx)
889         mov     %r12d, 12(%rbx)
890         xor     %rax, %rax
891 4:      / Restore registers
892         mov     1*8(%rsp), %rbx
893         mov     2*8(%rsp), %rbp
894         mov     3*8(%rsp), %r12
895         add     $[4*8], %rsp
901         /* EXPORT DELETE END */
896         ret

898         SET_SIZE(aes_decrypt_amd64)
```

*unchanged_portion_omitted_*

```
**********************************************************
   24758 Thu Jul 11 01:29:00 2013
new/usr/src/common/crypto/aes/amd64/aes_intel.s
first pass
**********************************************************
```

```
/*
 * ====================================================================
 * Written by Intel Corporation for the OpenSSL project to add support
 * for Intel AES-NI instructions. Rights for redistribution and usage
 * in source and binary forms are granted according to the OpenSSL
 * license.
 *
 *   Author: Huang Ying <ying.huang at intel dot com>
 *           Vinodh Gopal <vinodh.gopal at intel dot com>
 *           Kahraman Akdemir
 *
 * Intel AES-NI is a new set of Single Instruction Multiple Data (SIMD)
 * instructions that are going to be introduced in the next generation
 * of Intel processor, as of 2009. These instructions enable fast and
 * secure data encryption and decryption, using the Advanced Encryption
 * Standard (AES), defined by FIPS Publication number 197. The
 * architecture introduces six instructions that offer full hardware
 * support for AES. Four of them support high performance data
 * encryption and decryption, and the other two instructions support
 * the AES key expansion procedure.
 * ====================================================================
 */

/*
 * ====================================================================
 * Copyright (c) 1998-2008 The OpenSSL Project.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 *    software must display the following acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *    endorse or promote products derived from this software without
 *    prior written permission. For written permission, please contact
 *    openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 *    nor may "OpenSSL" appear in their names without prior written
 *    permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
 *    acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
```



```
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * ====================================================================
 */

/*
 * ====================================================================
 * OpenSolaris OS modifications
 *
 * This source originates as files aes-intel.S and eng_aesni_asm.pl, in
 * patches sent sent Dec. 9, 2008 and Dec. 24, 2008, respectively, by
 * Huang Ying of Intel to the openssl-dev mailing list under the subject
 * of "Add support to Intel AES-NI instruction set for x86_64 platform".
 *
 * This OpenSolaris version has these major changes from the original source:
 *
 * 1. Added OpenSolaris ENTRY_NP/SET_SIZE macros from
 * /usr/include/sys/asm_linkage.h, lint(1B) guards, and dummy C function
 * definitions for lint.
 * /usr/include/sys/asm_linkage.h, lint(1B) guards, EXPORT_DELETE START
 * and EXPORT_DELETE END markers, and dummy C function definitions for lint.
 *
 * 2. Formatted code, added comments, and added #includes and #defines.
 *
 * 3. If bit CR0.TS is set, clear and set the TS bit, after and before
 * calling kpreempt_disable() and kpreempt_enable().
 * If the TS bit is not set, Save and restore %xmm registers at the beginning
 * and end of function calls (%xmm* registers are not saved and restored by
 * during kernel thread preemption).
 *
 * 4. Renamed functions, reordered parameters, and changed return value
 * to match OpenSolaris:
 *
 * OpenSSL interface:
 *      int intel_AES_set_encrypt_key(const unsigned char *userKey,
 *              const int bits, AES_KEY *key);
 *      int intel_AES_set_decrypt_key(const unsigned char *userKey,
 *              const int bits, AES_KEY *key);
 *      Return values for above are non-zero on error, 0 on success.
 *
 *      void intel_AES_encrypt(const unsigned char *in, unsigned char *out,
 *              const AES_KEY *key);
 *      void intel_AES_decrypt(const unsigned char *in, unsigned char *out,
 *              const AES_KEY *key);
 *      typedef struct aes_key_st {
 *              unsigned int    rd_key[4 *(AES_MAXNR + 1)];
 *              int             rounds;
 *              unsigned int    pad[3];
 *      } AES_KEY;
 * Note: AES_LONG is undefined (that is, Intel uses 32-bit key schedules
 * (ks32) instead of 64-bit (ks64).
 * Number of rounds (aka round count) is at offset 240 of AES_KEY.
 *
 * OpenSolaris OS interface (#ifdefs removed for readability):
 *      int rijndael_key_setup_dec_intel(uint32_t rk[],
 *              const uint32_t cipherKey[], uint64_t keyBits);
 *      int rijndael_key_setup_enc_intel(uint32_t rk[],
 *              const uint32_t cipherKey[], uint64_t keyBits);
 *      Return values for above are 0 on error, number of rounds on success.
```

```
 126  *
 127  *      void aes_encrypt_intel(const aes_ks_t *ks, int Nr,
 128  *              const uint32_t pt[4], uint32_t ct[4]);
 129  *      void aes_decrypt_intel(const aes_ks_t *ks, int Nr,
 130  *              const uint32_t pt[4], uint32_t ct[4]);
 131  *      typedef union {uint64_t ks64[(MAX_AES_NR + 1) * 4];
 132  *              uint32_t ks32[(MAX_AES_NR + 1) * 4]; } aes_ks_t;
 133  *
 134  *      typedef union {
 135  *              uint32_t        ks32[((MAX_AES_NR) + 1) * (MAX_AES_NB)];
 136  *      } aes_ks_t;
 137  *      typedef struct aes_key {
 138  *              aes_ks_t        encr_ks, decr_ks;
 139  *              long double     align128;
 140  *              int             flags, nr, type;
 141  *      } aes_key_t;
 142  *
 143  * Note: ks is the AES key schedule, Nr is number of rounds, pt is plain text,
 144  * ct is crypto text, and MAX_AES_NR is 14.
 145  * For the x86 64-bit architecture, OpenSolaris OS uses ks32 instead of ks64.
 146  *
 147  * Note2: aes_ks_t must be aligned on a 0 mod 128 byte boundary.
 148  *
 149  * ====================================================================
 150  */

 152 #if defined(lint) || defined(__lint)

 154 #include <sys/types.h>

 156 /* ARGSUSED */
 157 void
 158 aes_encrypt_intel(const uint32_t rk[], int Nr, const uint32_t pt[4],
 159     uint32_t ct[4]) {
 160 }
_____unchanged_portion_omitted_


 180 #else   /* lint */

 182 #include <sys/asm_linkage.h>
 183 #include <sys/controlregs.h>
 184 #ifdef _KERNEL
 185 #include <sys/machprivregs.h>
 186 #endif

 188 #ifdef _KERNEL
 189         /*
 190          * Note: the CLTS macro clobbers P2 (%rsi) under i86xpv.  That is,
 191          * it calls HYPERVISOR_fpu_taskswitch() which modifies %rsi when it
 192          * uses it to pass P2 to syscall.
 193          * This also occurs with the STTS macro, but we don't care if
 194          * P2 (%rsi) is modified just before function exit.
 195          * The CLTS and STTS macros push and pop P1 (%rdi) already.
 196          */
 197 #ifdef __xpv
 198 #define PROTECTED_CLTS \
 199         push    %rsi; \
 200         CLTS; \
 201         pop     %rsi
 202 #else
 203 #define PROTECTED_CLTS \
 204         CLTS
 205 #endif  /* __xpv */

 207 #define CLEAR_TS_OR_PUSH_XMM0_XMM1(tmpreg) \
```



```
 208         push    %rbp; \
 209         mov     %rsp, %rbp; \
 210         movq    %cr0, tmpreg; \
 211         testq   $CR0_TS, tmpreg; \
 212         jnz     1f; \
 213         and     $-XMM_ALIGN, %rsp; \
 214         sub     $[XMM_SIZE * 2], %rsp; \
 215         movaps  %xmm0, 16(%rsp); \
 216         movaps  %xmm1, (%rsp); \
 217         jmp     2f; \
 218 1: \
 219         PROTECTED_CLTS; \
 220 2:

 222         /*
 223          * If CR0_TS was not set above, pop %xmm0 and %xmm1 off stack,
 224          * otherwise set CR0_TS.
 225          */
 226 #define SET_TS_OR_POP_XMM0_XMM1(tmpreg) \
 227         testq   $CR0_TS, tmpreg; \
 228         jnz     1f; \
 229         movaps  (%rsp), %xmm1; \
 230         movaps  16(%rsp), %xmm0; \
 231         jmp     2f; \
 232 1: \
 233         STTS(tmpreg); \
 234 2: \
 235         mov     %rbp, %rsp; \
 236         pop     %rbp

 238         /*
 239          * If CR0_TS is not set, align stack (with push %rbp) and push
 240          * %xmm0 - %xmm6 on stack, otherwise clear CR0_TS
 241          */
 242 #define CLEAR_TS_OR_PUSH_XMM0_TO_XMM6(tmpreg) \
 243         push    %rbp; \
 244         mov     %rsp, %rbp; \
 245         movq    %cr0, tmpreg; \
 246         testq   $CR0_TS, tmpreg; \
 247         jnz     1f; \
 248         and     $-XMM_ALIGN, %rsp; \
 249         sub     $[XMM_SIZE * 7], %rsp; \
 250         movaps  %xmm0, 96(%rsp); \
 251         movaps  %xmm1, 80(%rsp); \
 252         movaps  %xmm2, 64(%rsp); \
 253         movaps  %xmm3, 48(%rsp); \
 254         movaps  %xmm4, 32(%rsp); \
 255         movaps  %xmm5, 16(%rsp); \
 256         movaps  %xmm6, (%rsp); \
 257         jmp     2f; \
 258 1: \
 259         PROTECTED_CLTS; \
 260 2:


 263         /*
 264          * If CR0_TS was not set above, pop %xmm0 - %xmm6 off stack,
 265          * otherwise set CR0_TS.
 266          */
 267 #define SET_TS_OR_POP_XMM0_TO_XMM6(tmpreg) \
 268         testq   $CR0_TS, tmpreg; \
 269         jnz     1f; \
 270         movaps  (%rsp), %xmm6; \
 271         movaps  16(%rsp), %xmm5; \
 272         movaps  32(%rsp), %xmm4; \
 273         movaps  48(%rsp), %xmm3; \
```

```
274         movaps  64(%rsp), %xmm2; \
275         movaps  80(%rsp), %xmm1; \
276         movaps  96(%rsp), %xmm0; \
277         jmp     2f; \
278 1: \
279         STTS(tmpreg); \
280 2: \
281         mov     %rbp, %rsp; \
282         pop     %rbp


285 #else
286 #define PROTECTED_CLTS
287 #define CLEAR_TS_OR_PUSH_XMM0_XMM1(tmpreg)
288 #define SET_TS_OR_POP_XMM0_XMM1(tmpreg)
289 #define CLEAR_TS_OR_PUSH_XMM0_TO_XMM6(tmpreg)
290 #define SET_TS_OR_POP_XMM0_TO_XMM6(tmpreg)
291 #endif  /* _KERNEL */


294 /*
295  * _key_expansion_128(), * _key_expansion_192a(), _key_expansion_192b(),
296  * _key_expansion_256a(), _key_expansion_256b()
297  *
298  * Helper functions called by rijndael_key_setup_inc_intel().
299  * Also used indirectly by rijndael_key_setup_dec_intel().
300  *
301  * Input:
302  * %xmm0        User-provided cipher key
303  * %xmm1        Round constant
304  * Output:
305  * (%rcx)       AES key
306  */

308         /* EXPORT DELETE START */
308 .align  16
309 _key_expansion_128:
310 _key_expansion_256a:
311         pshufd  $0b11111111, %xmm1, %xmm1
312         shufps  $0b00010000, %xmm0, %xmm4
313         pxor    %xmm4, %xmm0
314         shufps  $0b10001100, %xmm0, %xmm4
315         pxor    %xmm4, %xmm0
316         pxor    %xmm1, %xmm0
317         movaps  %xmm0, (%rcx)
318         add     $0x10, %rcx
319         ret
320         SET_SIZE(_key_expansion_128)
______unchanged_portion_omitted_
381         /* EXPORT DELETE END */


382 /*
383  * rijndael_key_setup_enc_intel()
384  * Expand the cipher key into the encryption key schedule.
385  *
386  * For kernel code, caller is responsible for ensuring kpreempt_disable()
387  * has been called.  This is because %xmm registers are not saved/restored.
388  * Clear and set the CR0.TS bit on entry and exit, respectively,  if TS is set
389  * on entry.  Otherwise, if TS is not set, save and restore %xmm registers
390  * on the stack.
391  *
392  * OpenSolaris interface:
393  * int rijndael_key_setup_enc_intel(uint32_t rk[], const uint32_t cipherKey[],
394  *      uint64_t keyBits);
395  * Return value is 0 on error, number of rounds on success.
```



```
396  *
397  * Original Intel OpenSSL interface:
398  * int intel_AES_set_encrypt_key(const unsigned char *userKey,
399  *      const int bits, AES_KEY *key);
400  * Return value is non-zero on error, 0 on success.
401  */

403 #ifdef  OPENSSL_INTERFACE
404 #define rijndael_key_setup_enc_intel    intel_AES_set_encrypt_key
405 #define rijndael_key_setup_dec_intel    intel_AES_set_decrypt_key

407 #define USERCIPHERKEY           rdi     /* P1, 64 bits */
408 #define KEYSIZE32               esi     /* P2, 32 bits */
409 #define KEYSIZE64               rsi     /* P2, 64 bits */
410 #define AESKEY                  rdx     /* P3, 64 bits */

412 #else   /* OpenSolaris Interface */
413 #define AESKEY                  rdi     /* P1, 64 bits */
414 #define USERCIPHERKEY           rsi     /* P2, 64 bits */
415 #define KEYSIZE32               edx     /* P3, 32 bits */
416 #define KEYSIZE64               rdx     /* P3, 64 bits */
417 #endif  /* OPENSSL_INTERFACE */

419 #define ROUNDS32                KEYSIZE32       /* temp */
420 #define ROUNDS64                KEYSIZE64       /* temp */
421 #define ENDAESKEY               USERCIPHERKEY   /* temp */


424 ENTRY_NP(rijndael_key_setup_enc_intel)
427         /* EXPORT DELETE START */
425         CLEAR_TS_OR_PUSH_XMM0_TO_XMM6(%r10)

427         / NULL pointer sanity check
428         test    %USERCIPHERKEY, %USERCIPHERKEY
429         jz      .Lenc_key_invalid_param
430         test    %AESKEY, %AESKEY
431         jz      .Lenc_key_invalid_param

433         movups  (%USERCIPHERKEY), %xmm0 / user key (first 16 bytes)
434         movaps  %xmm0, (%AESKEY)
435         lea     0x10(%AESKEY), %rcx     / key addr
436         pxor    %xmm4, %xmm4            / xmm4 is assumed 0 in _key_expansion_x

438         cmp     $256, %KEYSIZE32
439         jnz     .Lenc_key192

441         / AES 256: 14 rounds in encryption key schedule
442 #ifdef OPENSSL_INTERFACE
443         mov     $14, %ROUNDS32
444         movl    %ROUNDS32, 240(%AESKEY)         / key.rounds = 14
445 #endif  /* OPENSSL_INTERFACE */

447         movups  0x10(%USERCIPHERKEY), %xmm2     / other user key (2nd 16 bytes)
448         movaps  %xmm2, (%rcx)
449         add     $0x10, %rcx

451         aeskeygenassist $0x1, %xmm2, %xmm1      / expand the key
452         call    _key_expansion_256a
453         aeskeygenassist $0x1, %xmm0, %xmm1
454         call    _key_expansion_256b
455         aeskeygenassist $0x2, %xmm2, %xmm1      / expand the key
456         call    _key_expansion_256a
457         aeskeygenassist $0x2, %xmm0, %xmm1
458         call    _key_expansion_256b
459         aeskeygenassist $0x4, %xmm2, %xmm1      / expand the key
460         call    _key_expansion_256a
```

```
461         aeskeygenassist $0x4, %xmm0, %xmm1
462         call    _key_expansion_256b
463         aeskeygenassist $0x8, %xmm2, %xmm1      / expand the key
464         call    _key_expansion_256a
465         aeskeygenassist $0x8, %xmm0, %xmm1
466         call    _key_expansion_256b
467         aeskeygenassist $0x10, %xmm2, %xmm1     / expand the key
468         call    _key_expansion_256a
469         aeskeygenassist $0x10, %xmm0, %xmm1
470         call    _key_expansion_256b
471         aeskeygenassist $0x20, %xmm2, %xmm1     / expand the key
472         call    _key_expansion_256a
473         aeskeygenassist $0x20, %xmm0, %xmm1
474         call    _key_expansion_256b
475         aeskeygenassist $0x40, %xmm2, %xmm1     / expand the key
476         call    _key_expansion_256a

478         SET_TS_OR_POP_XMM0_TO_XMM6(%r10)
479 #ifdef  OPENSSL_INTERFACE
480         xor     %rax, %rax                      / return 0 (OK)
481 #else   /* Open Solaris Interface */
482         mov     $14, %rax                       / return # rounds = 14
483 #endif
484         ret

486 .align 4
487 .Lenc_key192:
488         cmp     $192, %KEYSIZE32
489         jnz     .Lenc_key128

491         / AES 192: 12 rounds in encryption key schedule
492 #ifdef OPENSSL_INTERFACE
493         mov     $12, %ROUNDS32
494         movl    %ROUNDS32, 240(%AESKEY) / key.rounds = 12
495 #endif  /* OPENSSL_INTERFACE */

497         movq    0x10(%USERCIPHERKEY), %xmm2     / other user key
498         aeskeygenassist $0x1, %xmm2, %xmm1      / expand the key
499         call    _key_expansion_192a
500         aeskeygenassist $0x2, %xmm2, %xmm1      / expand the key
501         call    _key_expansion_192b
502         aeskeygenassist $0x4, %xmm2, %xmm1      / expand the key
503         call    _key_expansion_192a
504         aeskeygenassist $0x8, %xmm2, %xmm1      / expand the key
505         call    _key_expansion_192b
506         aeskeygenassist $0x10, %xmm2, %xmm1     / expand the key
507         call    _key_expansion_192a
508         aeskeygenassist $0x20, %xmm2, %xmm1     / expand the key
509         call    _key_expansion_192b
510         aeskeygenassist $0x40, %xmm2, %xmm1     / expand the key
511         call    _key_expansion_192a
512         aeskeygenassist $0x80, %xmm2, %xmm1     / expand the key
513         call    _key_expansion_192b

515         SET_TS_OR_POP_XMM0_TO_XMM6(%r10)
516 #ifdef  OPENSSL_INTERFACE
517         xor     %rax, %rax                      / return 0 (OK)
518 #else   /* OpenSolaris Interface */
519         mov     $12, %rax                       / return # rounds = 12
520 #endif
521         ret

523 .align 4
524 .Lenc_key128:
525         cmp $128, %KEYSIZE32
526         jnz .Lenc_key_invalid_key_bits
```



```
528         / AES 128: 10 rounds in encryption key schedule
529 #ifdef OPENSSL_INTERFACE
530         mov     $10, %ROUNDS32
531         movl    %ROUNDS32, 240(%AESKEY)         / key.rounds = 10
532 #endif  /* OPENSSL_INTERFACE */

534         aeskeygenassist $0x1, %xmm0, %xmm1      / expand the key
535         call    _key_expansion_128
536         aeskeygenassist $0x2, %xmm0, %xmm1      / expand the key
537         call    _key_expansion_128
538         aeskeygenassist $0x4, %xmm0, %xmm1      / expand the key
539         call    _key_expansion_128
540         aeskeygenassist $0x8, %xmm0, %xmm1      / expand the key
541         call    _key_expansion_128
542         aeskeygenassist $0x10, %xmm0, %xmm1     / expand the key
543         call    _key_expansion_128
544         aeskeygenassist $0x20, %xmm0, %xmm1     / expand the key
545         call    _key_expansion_128
546         aeskeygenassist $0x40, %xmm0, %xmm1     / expand the key
547         call    _key_expansion_128
548         aeskeygenassist $0x80, %xmm0, %xmm1     / expand the key
549         call    _key_expansion_128
550         aeskeygenassist $0x1b, %xmm0, %xmm1     / expand the key
551         call    _key_expansion_128
552         aeskeygenassist $0x36, %xmm0, %xmm1     / expand the key
553         call    _key_expansion_128

555         SET_TS_OR_POP_XMM0_TO_XMM6(%r10)
556 #ifdef  OPENSSL_INTERFACE
557         xor     %rax, %rax                      / return 0 (OK)
558 #else   /* OpenSolaris Interface */
559         mov     $10, %rax                       / return # rounds = 10
560 #endif
561         ret

563 .Lenc_key_invalid_param:
564 #ifdef  OPENSSL_INTERFACE
565         SET_TS_OR_POP_XMM0_TO_XMM6(%r10)
566         mov     $-1, %rax       / user key or AES key pointer is NULL
567         ret
568 #else
569         /* FALLTHROUGH */
570 #endif  /* OPENSSL_INTERFACE */

572 .Lenc_key_invalid_key_bits:
573         SET_TS_OR_POP_XMM0_TO_XMM6(%r10)
574 #ifdef  OPENSSL_INTERFACE
575         mov     $-2, %rax       / keysize is invalid
576 #else   /* Open Solaris Interface */
577         xor     %rax, %rax      / a key pointer is NULL or invalid keysize
578 #endif  /* OPENSSL_INTERFACE */

583         /* EXPORT DELETE END */
580         ret
581         SET_SIZE(rijndael_key_setup_enc_intel)


584 /*
585  * rijndael_key_setup_dec_intel()
586  * Expand the cipher key into the decryption key schedule.
587  *
588  * For kernel code, caller is responsible for ensuring kpreempt_disable()
589  * has been called.  This is because %xmm registers are not saved/restored.
590  * Clear and set the CR0.TS bit on entry and exit, respectively,  if TS is set
591  * on entry.  Otherwise, if TS is not set, save and restore %xmm registers
```

```
 592  * on the stack.
 593  *
 594  * OpenSolaris interface:
 595  * int rijndael_key_setup_dec_intel(uint32_t rk[], const uint32_t cipherKey[],
 596  *      uint64_t keyBits);
 597  * Return value is 0 on error, number of rounds on success.
 598  * P1->P2, P2->P3, P3->P1
 599  *
 600  * Original Intel OpenSSL interface:
 601  * int intel_AES_set_decrypt_key(const unsigned char *userKey,
 602  *      const int bits, AES_KEY *key);
 603  * Return value is non-zero on error, 0 on success.
 604  */
 605 ENTRY_NP(rijndael_key_setup_dec_intel)
 610         /* EXPORT DELETE START */
 606         / Generate round keys used for encryption
 607         call    rijndael_key_setup_enc_intel
 608         test    %rax, %rax
 609 #ifdef  OPENSSL_INTERFACE
 610         jnz     .Ldec_key_exit  / Failed if returned non-0
 611 #else   /* OpenSolaris Interface */
 612         jz      .Ldec_key_exit  / Failed if returned 0
 613 #endif  /* OPENSSL_INTERFACE */

 615         CLEAR_TS_OR_PUSH_XMM0_XMM1(%r10)

 617         /*
 618          * Convert round keys used for encryption
 619          * to a form usable for decryption
 620          */
 621 #ifndef OPENSSL_INTERFACE               /* OpenSolaris Interface */
 622         mov     %rax, %ROUNDS64         / set # rounds (10, 12, or 14)
 623                                         / (already set for OpenSSL)
 624 #endif

 626         lea     0x10(%AESKEY), %rcx     / key addr
 627         shl     $4, %ROUNDS32
 628         add     %AESKEY, %ROUNDS64
 629         mov     %ROUNDS64, %ENDAESKEY

 631 .align 4
 632 .Ldec_key_reorder_loop:
 633         movaps  (%AESKEY), %xmm0
 634         movaps  (%ROUNDS64), %xmm1
 635         movaps  %xmm0, (%ROUNDS64)
 636         movaps  %xmm1, (%AESKEY)
 637         lea     0x10(%AESKEY), %AESKEY
 638         lea     -0x10(%ROUNDS64), %ROUNDS64
 639         cmp     %AESKEY, %ROUNDS64
 640         ja      .Ldec_key_reorder_loop

 642 .align 4
 643 .Ldec_key_inv_loop:
 644         movaps  (%rcx), %xmm0
 645         / Convert an encryption round key to a form usable for decryption
 646         / with the "AES Inverse Mix Columns" instruction
 647         aesimc  %xmm0, %xmm1
 648         movaps  %xmm1, (%rcx)
 649         lea     0x10(%rcx), %rcx
 650         cmp     %ENDAESKEY, %rcx
 651         jnz     .Ldec_key_inv_loop

 653         SET_TS_OR_POP_XMM0_XMM1(%r10)

 655 .Ldec_key_exit:
 656         / OpenSolaris: rax = # rounds (10, 12, or 14) or 0 for error
```



```
 657         / OpenSSL: rax = 0 for OK, or non-zero for error
 663         /* EXPORT DELETE END */
 658         ret
 659         SET_SIZE(rijndael_key_setup_dec_intel)


 662 /*
 663  * aes_encrypt_intel()
 664  * Encrypt a single block (in and out can overlap).
 665  *
 666  * For kernel code, caller is responsible for ensuring kpreempt_disable()
 667  * has been called.  This is because %xmm registers are not saved/restored.
 668  * Clear and set the CR0.TS bit on entry and exit, respectively,  if TS is set
 669  * on entry.  Otherwise, if TS is not set, save and restore %xmm registers
 670  * on the stack.
 671  *
 672  * Temporary register usage:
 673  * %xmm0        State
 674  * %xmm1        Key
 675  *
 676  * Original OpenSolaris Interface:
 677  * void aes_encrypt_intel(const aes_ks_t *ks, int Nr,
 678  *      const uint32_t pt[4], uint32_t ct[4])
 679  *
 680  * Original Intel OpenSSL Interface:
 681  * void intel_AES_encrypt(const unsigned char *in, unsigned char *out,
 682  *      const AES_KEY *key)
 683  */

 685 #ifdef  OPENSSL_INTERFACE
 686 #define aes_encrypt_intel       intel_AES_encrypt
 687 #define aes_decrypt_intel       intel_AES_decrypt

 689 #define INP             rdi     /* P1, 64 bits */
 690 #define OUTP            rsi     /* P2, 64 bits */
 691 #define KEYP            rdx     /* P3, 64 bits */

 693 /* No NROUNDS parameter--offset 240 from KEYP saved in %ecx:  */
 694 #define NROUNDS32       ecx     /* temporary, 32 bits */
 695 #define NROUNDS         cl      /* temporary,  8 bits */

 697 #else   /* OpenSolaris Interface */
 698 #define KEYP            rdi     /* P1, 64 bits */
 699 #define NROUNDS         esi     /* P2, 32 bits */
 700 #define INP             rdx     /* P3, 64 bits */
 701 #define OUTP            rcx     /* P4, 64 bits */
 702 #endif  /* OPENSSL_INTERFACE */

 704 #define STATE           xmm0    /* temporary, 128 bits */
 705 #define KEY             xmm1    /* temporary, 128 bits */

 707 ENTRY_NP(aes_encrypt_intel)
 714         /* EXPORT DELETE START */
 708         CLEAR_TS_OR_PUSH_XMM0_XMM1(%r10)

 710         movups  (%INP), %STATE                  / input
 711         movaps  (%KEYP), %KEY                   / key
 712 #ifdef  OPENSSL_INTERFACE
 713         mov     240(%KEYP), %NROUNDS32          / round count
 714 #else   /* OpenSolaris Interface */
 715         /* Round count is already present as P2 in %rsi/%esi */
 716 #endif  /* OPENSSL_INTERFACE */

 718         pxor    %KEY, %STATE                    / round 0
 719         lea     0x30(%KEYP), %KEYP
 720         cmp     $12, %NROUNDS
```

```
721         jb      .Lenc128
722         lea     0x20(%KEYP), %KEYP
723         je      .Lenc192

725         / AES 256
726         lea     0x20(%KEYP), %KEYP
727         movaps  -0x60(%KEYP), %KEY
728         aesenc  %KEY, %STATE
729         movaps  -0x50(%KEYP), %KEY
730         aesenc  %KEY, %STATE

732 .align 4
733 .Lenc192:
734         / AES 192 and 256
735         movaps  -0x40(%KEYP), %KEY
736         aesenc  %KEY, %STATE
737         movaps  -0x30(%KEYP), %KEY
738         aesenc  %KEY, %STATE

740 .align 4
741 .Lenc128:
742         / AES 128, 192, and 256
743         movaps  -0x20(%KEYP), %KEY
744         aesenc  %KEY, %STATE
745         movaps  -0x10(%KEYP), %KEY
746         aesenc  %KEY, %STATE
747         movaps  (%KEYP), %KEY
748         aesenc  %KEY, %STATE
749         movaps  0x10(%KEYP), %KEY
750         aesenc  %KEY, %STATE
751         movaps  0x20(%KEYP), %KEY
752         aesenc  %KEY, %STATE
753         movaps  0x30(%KEYP), %KEY
754         aesenc  %KEY, %STATE
755         movaps  0x40(%KEYP), %KEY
756         aesenc  %KEY, %STATE
757         movaps  0x50(%KEYP), %KEY
758         aesenc  %KEY, %STATE
759         movaps  0x60(%KEYP), %KEY
760         aesenc  %KEY, %STATE
761         movaps  0x70(%KEYP), %KEY
762         aesenclast      %KEY, %STATE            / last round
763         movups  %STATE, (%OUTP)                 / output

765         SET_TS_OR_POP_XMM0_XMM1(%r10)
773         /* EXPORT DELETE END */
766         ret
767         SET_SIZE(aes_encrypt_intel)


770 /*
771  * aes_decrypt_intel()
772  * Decrypt a single block (in and out can overlap).
773  *
774  * For kernel code, caller is responsible for ensuring kpreempt_disable()
775  * has been called.  This is because %xmm registers are not saved/restored.
776  * Clear and set the CR0.TS bit on entry and exit, respectively,  if TS is set
777  * on entry.  Otherwise, if TS is not set, save and restore %xmm registers
778  * on the stack.
779  *
780  * Temporary register usage:
781  * %xmm0        State
782  * %xmm1        Key
783  *
784  * Original OpenSolaris Interface:
785  * void aes_decrypt_intel(const aes_ks_t *ks, int Nr,
```



```
786  *      const uint32_t pt[4], uint32_t ct[4])/
787  *
788  * Original Intel OpenSSL Interface:
789  * void intel_AES_decrypt(const unsigned char *in, unsigned char *out,
790  *      const AES_KEY *key);
791  */
792 ENTRY_NP(aes_decrypt_intel)
801         /* EXPORT DELETE START */
793         CLEAR_TS_OR_PUSH_XMM0_XMM1(%r10)

795         movups  (%INP), %STATE                  / input
796         movaps  (%KEYP), %KEY                   / key
797 #ifdef  OPENSSL_INTERFACE
798         mov     240(%KEYP), %NROUNDS32          / round count
799 #else   /* OpenSolaris Interface */
800         /* Round count is already present as P2 in %rsi/%esi */
801 #endif  /* OPENSSL_INTERFACE */

803         pxor    %KEY, %STATE                    / round 0
804         lea     0x30(%KEYP), %KEYP
805         cmp     $12, %NROUNDS
806         jb      .Ldec128
807         lea     0x20(%KEYP), %KEYP
808         je      .Ldec192

810         / AES 256
811         lea     0x20(%KEYP), %KEYP
812         movaps  -0x60(%KEYP), %KEY
813         aesdec  %KEY, %STATE
814         movaps  -0x50(%KEYP), %KEY
815         aesdec  %KEY, %STATE

817 .align 4
818 .Ldec192:
819         / AES 192 and 256
820         movaps  -0x40(%KEYP), %KEY
821         aesdec  %KEY, %STATE
822         movaps  -0x30(%KEYP), %KEY
823         aesdec  %KEY, %STATE

825 .align 4
826 .Ldec128:
827         / AES 128, 192, and 256
828         movaps  -0x20(%KEYP), %KEY
829         aesdec  %KEY, %STATE
830         movaps  -0x10(%KEYP), %KEY
831         aesdec  %KEY, %STATE
832         movaps  (%KEYP), %KEY
833         aesdec  %KEY, %STATE
834         movaps  0x10(%KEYP), %KEY
835         aesdec  %KEY, %STATE
836         movaps  0x20(%KEYP), %KEY
837         aesdec  %KEY, %STATE
838         movaps  0x30(%KEYP), %KEY
839         aesdec  %KEY, %STATE
840         movaps  0x40(%KEYP), %KEY
841         aesdec  %KEY, %STATE
842         movaps  0x50(%KEYP), %KEY
843         aesdec  %KEY, %STATE
844         movaps  0x60(%KEYP), %KEY
845         aesdec  %KEY, %STATE
846         movaps  0x70(%KEYP), %KEY
847         aesdeclast      %KEY, %STATE            / last round
848         movups  %STATE, (%OUTP)                 / output

850         SET_TS_OR_POP_XMM0_XMM1(%r10)
```

```
 851         ret
 861         /* EXPORT DELETE END */
 852         SET_SIZE(aes_decrypt_intel)
```

unchanged_portion_omitted_

```
**********************************************************
   89905 Thu Jul 11 01:29:00 2013
new/usr/src/common/crypto/aes/sun4u/aes_crypt_asm.s
first pass
**********************************************************
______unchanged_portion_omitted_

  70 #else   /* lint || __lint */

  72         .section        ".text",#alloc,#execinstr
  73         .file   "aes_crypt_asm.s"

  75         .register       %g2,#scratch
  76         .register       %g3,#scratch

  78         .section        ".text",#alloc
  79         .align  8192

  81 /* EXPORT DELETE START */

  81 !
  82 ! CONSTANT POOL
  83 !
  84 Te0:
  85         .word   1584
  86         .word   1661148456
  87         .word   1984
  88         .word   2080629792
  89         .word   1904
  90         .word   1996733640
  91         .word   1968
  92         .word   2063850600
  93         .word   2040
  94         .word   -234385304
  95         .word   1712
  96         .word   1795382760
  97         .word   1776
  98         .word   1862499720
  99         .word   1160
 100         .word   -989451616
 101         .word   768
 102         .word   805405312
 103         .word   16
 104         .word   16779288
 105         .word   1648
 106         .word   1728265544
 107         .word   688
 108         .word   721509352
 109         .word   1848
 110         .word   -33034040
 111         .word   1448
 112         .word   -687424752
 113         .word   616
 114         .word   -1425711312
 115         .word   1888
 116         .word   1979954384
 117         .word   1144
 118         .word   -905555416
 119         .word   248
 120         .word   -2113661720
 121         .word   1096
 122         .word   -922334720
 123         .word   2000
 124         .word   2097409080
 125         .word   1912
 126         .word   -100151128
```



```
 127         .word   1424
 128         .word   1493356376
 129         .word   1136
 130         .word   1191329352
 131         .word   2008
 132         .word   -267943848
 133         .word   520
 134         .word   -1392152736
 135         .word   1432
 136         .word   -737762504
 137         .word   760
 138         .word   -1576724504
 139         .word   552
 140         .word   -1358594224
 141         .word   280
 142         .word   -1677400584
 143         .word   664
 144         .word   -1543166024
 145         .word   1824
 146         .word   1912837296
 147         .word   1240
 148         .word   -1073347880
 149         .word   936
 150         .word   -1224360432
 151         .word   1800
 152         .word   -49813280
 153         .word   488
 154         .word   -1828414096
 155         .word   608
 156         .word   637612880
 157         .word   864
 158         .word   906080976
 159         .word   1008
 160         .word   1057094152
 161         .word   1960
 162         .word   -150489072
 163         .word   1048
 164         .word   -871996808
 165         .word   832
 166         .word   872522464
 167         .word   648
 168         .word   -1526386784
 169         .word   1672
 170         .word   -452515424
 171         .word   1992
 172         .word   -251164608
 173         .word   1808
 174         .word   1896058008
 175         .word   1368
 176         .word   -670645352
 177         .word   784
 178         .word   822184600
 179         .word   336
 180         .word   352365048
 181         .word   64
 182         .word   67117152
 183         .word   1192
 184         .word   -955893104
 185         .word   560
 186         .word   587275048
 187         .word   1256
 188         .word   -1023010064
 189         .word   384
 190         .word   402702656
 191         .word   440
 192         .word   -1778076408
```

```
193         .word   80
194         .word   83896440
195         .word   376
196         .word   -1710959192
197         .word   112
198         .word   117454920
199         .word   288
200         .word   302027184
201         .word   216
202         .word   -2147220264
203         .word   1784
204         .word   -502853144
205         .word   1640
206         .word   -351839952
207         .word   624
208         .word   654392136
209         .word   1016
210         .word   -1308256664
211         .word   1872
212         .word   1963175160
213         .word   144
214         .word   151013592
215         .word   232
216         .word   -2096882448
217         .word   704
218         .word   738288544
219         .word   416
220         .word   436261232
221         .word   432
222         .word   453040488
223         .word   1760
224         .word   1845720464
225         .word   1440
226         .word   1510135664
227         .word   728
228         .word   -1610283048
229         .word   1312
230         .word   1375901616
231         .word   944
232         .word   989977192
233         .word   1464
234         .word   -704204024
235         .word   1000
236         .word   -1291477392
237         .word   656
238         .word   687950808
239         .word   1768
240         .word   -486073872
241         .word   752
242         .word   788626312
243         .word   152
244         .word   -2080103240
245         .word   1328
246         .word   1392680872
247         .word   1480
248         .word   -788100288
249         .word   0
250         .word   0
251         .word   1544
252         .word   -318281376
253         .word   512
254         .word   536937216
255         .word   1816
256         .word   -66592520
257         .word   968
258         .word   -1325035968
```



```
259         .word   1456
260         .word   1526914920
261         .word   1696
262         .word   1778603504
263         .word   1128
264         .word   -888776144
265         .word   824
266         .word   -1106905400
267         .word   912
268         .word   956418648
269         .word   1184
270         .word   1241667312
271         .word   1216
272         .word   1275225760
273         .word   1408
274         .word   1476577088
275         .word   1064
276         .word   -821659056
277         .word   1496
278         .word   -804879528
279         .word   1576
280         .word   -284722864
281         .word   632
282         .word   -1442490584
283         .word   1896
284         .word   -83371856
285         .word   1072
286         .word   1124212264
287         .word   1232
288         .word   1292005048
289         .word   816
290         .word   855743144
291         .word   136
292         .word   -2063324000
293         .word   1104
294         .word   1157770872
295         .word   1864
296         .word   -116930432
297         .word   32
298         .word   33558576
299         .word   2032
300         .word   2130967560
301         .word   1280
302         .word   1342343040
303         .word   960
304         .word   1006756384
305         .word   296
306         .word   -1627062832
307         .word   600
308         .word   -1476049128
309         .word   1296
310         .word   1359122328
311         .word   744
312         .word   -1559945232
313         .word   1024
314         .word   1073874432
315         .word   40
316         .word   -1895531440
317         .word   504
318         .word   -1845193368
319         .word   264
320         .word   -1660621344
321         .word   896
322         .word   939639360
323         .word   1928
324         .word   -184047584
```

```
325         .word   792
326         .word   -1140463880
327         .word   952
328         .word   -1241139704
329         .word   1400
330         .word   -637086808
331         .word   528
332         .word   553716504
333         .word   256
334         .word   268468608
335         .word   1832
336         .word   -16254768
337         .word   2024
338         .word   -217606032
339         .word   1528
340         .word   -771320984
341         .word   1032
342         .word   -855217568
343         .word   192
344         .word   201351328
345         .word   304
346         .word   318806440
347         .word   1560
348         .word   -335060616
349         .word   1520
350         .word   1594031880
351         .word   424
352         .word   -1761297136
353         .word   1088
354         .word   1140991584
355         .word   368
356         .word   385923528
357         .word   1176
358         .word   -1006230856
359         .word   680
360         .word   -1492828272
361         .word   2016
362         .word   2114188304
363         .word   976
364         .word   1023535672
365         .word   1600
366         .word   1677927776
367         .word   1488
368         .word   1560473400
369         .word   400
370         .word   419481944
371         .word   1840
372         .word   1929616552
373         .word   1536
374         .word   1610810624
375         .word   200
376         .word   -2130441024
377         .word   1264
378         .word   1325563528
379         .word   1304
380         .word   -603528200
381         .word   544
382         .word   570495792
383         .word   672
384         .word   704730096
385         .word   472
386         .word   -1878751912
387         .word   88
388         .word   -2012986344
389         .word   1120
390         .word   1174550096
```



```
391         .word   1592
392         .word   -301502136
393         .word   856
394         .word   -1207581032
395         .word   320
396         .word   335585760
397         .word   1336
398         .word   -569969720
399         .word   1504
400         .word   1577252624
401         .word   176
402         .word   184572136
403         .word   1384
404         .word   -620307536
405         .word   1752
406         .word   -536411688
407         .word   800
408         .word   838963888
409         .word   928
410         .word   973197936
411         .word   160
412         .word   167792880
413         .word   1168
414         .word   1224888024
415         .word   96
416         .word   100675664
417         .word   576
418         .word   604054368
419         .word   1472
420         .word   1543694112
421         .word   1272
422         .word   -1039789336
423         .word   1512
424         .word   -754541712
425         .word   536
426         .word   -1408931976
427         .word   1568
428         .word   1644369200
429         .word   456
430         .word   -1861972672
431         .word   392
432         .word   -1794855648
433         .word   1688
434         .word   -469294664
435         .word   1936
436         .word   2030292056
437         .word   1704
438         .word   -418956912
439         .word   1112
440         .word   -939113960
441         .word   880
442         .word   922860232
443         .word   1744
444         .word   1828941240
445         .word   8
446         .word   -1929089952
447         .word   1416
448         .word   -720983264
449         .word   1248
450         .word   1308784272
451         .word   584
452         .word   -1459269888
453         .word   1728
454         .word   1812161952
455         .word   1376
456         .word   1443018704
```

```
457         .word   1944
458         .word   -200826824
459         .word   1656
460         .word   -368619224
461         .word   1616
462         .word   1694707064
463         .word   1952
464         .word   2047071344
465         .word   568
466         .word   -1375373496
467         .word   128
468         .word   134234304
469         .word   888
470         .word   -1174022488
471         .word   1920
472         .word   2013512768
473         .word   592
474         .word   620833656
475         .word   736
476         .word   771847056
477         .word   448
478         .word   469819680
479         .word   696
480         .word   -1509607544
481         .word   920
482         .word   -1274698184
483         .word   1208
484         .word   -972672376
485         .word   1624
486         .word   -402177768
487         .word   1288
488         .word   -586748960
489         .word   1856
490         .word   1946395872
491         .word   496
492         .word   520157448
493         .word   1200
494         .word   1258446568
495         .word   776
496         .word   -1123684640
497         .word   104
498         .word   -1962648528
499         .word   120
500         .word   -1979427800
501         .word   1792
502         .word   1879278720
503         .word   992
504         .word   1040314896
505         .word   904
506         .word   -1257918944
507         .word   1632
508         .word   1711486288
509         .word   1152
510         .word   1208108736
511         .word   48
512         .word   50337832
513         .word   1976
514         .word   -167268344
515         .word   224
516         .word   234909840
517         .word   1552
518         .word   1627589912
519         .word   848
520         .word   889301752
521         .word   1392
522         .word   1459797960
```



```
523         .word   840
524         .word   -1190801792
525         .word   184
526         .word   -2046544760
527         .word   1224
528         .word   -1056568640
529         .word   464
530         .word   486598968
531         .word   312
532         .word   -1643842104
533         .word   1736
534         .word   -519632448
535         .word   1880
536         .word   -133709672
537         .word   344
538         .word   -1744517736
539         .word   272
540         .word   285247896
541         .word   1680
542         .word   1761824216
543         .word   1352
544         .word   -653866112
545         .word   56
546         .word   -1912310712
547         .word   408
548         .word   -1811634888
549         .word   360
550         .word   -1694179920
551         .word   480
552         .word   503378192
553         .word   168
554         .word   -2029765488
555         .word   1608
556         .word   -385398528
557         .word   1080
558         .word   -838438328
559         .word   1360
560         .word   1426239480
561         .word   640
562         .word   671171520
563         .word   1320
564         .word   -553190448
565         .word   24
566         .word   -1945869192
567         .word   712
568         .word   -1593503808
569         .word   72
570         .word   -1996207104
571         .word   208
572         .word   218130616
573         .word   808
574         .word   -1090126128
575         .word   1720
576         .word   -435736184
577         .word   1056
578         .word   1107433008
579         .word   1664
580         .word   1745044928
581         .word   1040
582         .word   1090653720
583         .word   328
584         .word   -1727738496
585         .word   720
586         .word   755067832
587         .word   240
588         .word   251689096
```

```
589         .word   984
590         .word   -1341815208
591         .word   1344
592         .word   1409460192
593         .word   872
594         .word   -1157243216
595         .word   352
596         .word   369144272
597         .type   Te0,#object
598         .size   Te0,2048
599 !
600 ! CONSTANT POOL
601 !
602 Te1:
603         .word   1320
604         .word   -972874984
605         .word   1056
606         .word   -133962784
607         .word   1224
608         .word   -301745224
609         .word   1128
610         .word   -167519272
611         .word   104
612         .word   -16279664
613         .word   1512
614         .word   -704423080
615         .word   1416
616         .word   -570197128
617         .word   672
618         .word   -1861865944
619         .word   640
620         .word   1610711424
621         .word   24
622         .word   33556488
623         .word   1352
624         .word   -838649032
625         .word   1000
626         .word   1442928984
627         .word   200
628         .word   -418908176
629         .word   784
630         .word   -1257849160
631         .word   1840
632         .word   1292197208
633         .word   1232
634         .word   -335301712
635         .word   552
636         .word   -1895410096
637         .word   1256
638         .word   520360976
639         .word   512
640         .word   -1996075448
641         .word   1080
642         .word   -100406296
643         .word   168
644         .word   -284698672
645         .word   1880
646         .word   -1308439864
647         .word   1608
648         .word   -1912456648
649         .word   88
650         .word   -83392640
651         .word   1888
652         .word   1090874728
653         .word   824
654         .word   -1291409760
```



```
655         .word   2024
656         .word   1594168592
657         .word   1872
658         .word   1157987704
659         .word   1528
660         .word   587523296
661         .word   1976
662         .word   1392846112
663         .word   1200
664         .word   -469527664
665         .word   728
666         .word   -1694104064
667         .word   1552
668         .word   1963310520
669         .word   224
670         .word   -519573528
671         .word   1392
672         .word   1023712408
673         .word   848
674         .word   1275146544
675         .word   720
676         .word   1812050352
677         .word   520
678         .word   2114058744
679         .word   16
680         .word   -184041544
681         .word   632
682         .word   -2096732576
683         .word   736
684         .word   1744937376
685         .word   1952
686         .word   1359293736
687         .word   416
688         .word   -788058328
689         .word   64
690         .word   -116945016
691         .word   1176
692         .word   -503084152
693         .word   920
694         .word   -1425619264
695         .word   664
696         .word   1644267912
697         .word   504
698         .word   704686248
699         .word   96
700         .word   134225952
701         .word   656
702         .word   -1794752968
703         .word   808
704         .word   1174477080
705         .word   752
706         .word   -1660543464
707         .word   320
708         .word   805355712
709         .word   1288
710         .word   923055280
711         .word   120
712         .word   167782440
713         .word   1448
714         .word   788845776
715         .word   72
716         .word   234895416
717         .word   432
718         .word   604016784
719         .word   1240
720         .word   453248000
```

```
721         .word   488
722         .word   -553183472
723         .word   304
724         .word   -855154856
725         .word   840
726         .word   1308703032
727         .word   1640
728         .word   2131072400
729         .word   1272
730         .word   -368858200
731         .word   216
732         .word   302008392
733         .word   1264
734         .word   486808600
735         .word   928
736         .word   1476485472
737         .word   368
738         .word   872468688
739         .word   360
740         .word   906025176
741         .word   1424
742         .word   -603753616
743         .word   1904
744         .word   -1274883376
745         .word   2008
746         .word   1527055616
747         .word   1968
748         .word   -1543335280
749         .word   616
750         .word   1979832792
751         .word   776
752         .word   -1224296784
753         .word   1648
754         .word   2097520024
755         .word   984
756         .word   1375816008
757         .word   496
758         .word   -586735848
759         .word   904
760         .word   1577154936
761         .word   1208
762         .word   319038496
763         .word   1960
764         .word   -1509778792
765         .word   832
766         .word   -1190752632
767         .word   0
768         .word   0
769         .word   352
770         .word   -1056477336
771         .word   768
772         .word   1073807616
773         .word   248
774         .word   -486021152
775         .word   1600
776         .word   2030407048
777         .word   1896
778         .word   -1241326888
779         .word   1520
780         .word   -737979568
781         .word   560
782         .word   -1928962472
783         .word   1736
784         .word   1728443888
785         .word   600
786         .word   1912719816
```



```
787         .word   1776
788         .word   -1811787184
789         .word   1696
790         .word   -1744674208
791         .word   1856
792         .word   -1341996352
793         .word   592
794         .word   -2063171976
795         .word   856
796         .word   -1157200256
797         .word   336
798         .word   -989364360
799         .word   1832
800         .word   1325749584
801         .word   176
802         .word   -318251048
803         .word   1576
804         .word   -2046682600
805         .word   1720
806         .word   -1711117720
807         .word   680
808         .word   1711380888
809         .word   1184
810         .word   285486120
811         .word   1656
812         .word   -1979569624
813         .word   128
814         .word   -385364024
815         .word   48
816         .word   67112976
817         .word   1032
818         .word   -33293320
819         .word   1920
820         .word   -1610448256
821         .word   544
822         .word   2013389280
823         .word   1488
824         .word   621083896
825         .word   1816
826         .word   1258636608
827         .word   1944
828         .word   -1576891768
829         .word   2032
830         .word   1560616216
831         .word   1536
832         .word   -2147352064
833         .word   1104
834         .word   84180088
835         .word   1384
836         .word   1057264784
837         .word   1504
838         .word   553970920
839         .word   576
840         .word   1879163328
841         .word   32
842         .word   -251154520
843         .word   1784
844         .word   1661330912
845         .word   1544
846         .word   1996862896
847         .word   936
848         .word   -1358506288
849         .word   792
850         .word   1107364104
851         .word   384
852         .word   536903808
```

```
853         .word   208
854         .word   -452460552
855         .word   112
856         .word   -49832040
857         .word   872
858         .word   -1090087280
859         .word   608
860         .word   -2130284952
861         .word   160
862         .word   402677856
863         .word   424
864         .word   637573272
865         .word   376
866         .word   -1022924960
867         .word   1800
868         .word   -1107100936
869         .word   1296
870         .word   889502904
871         .word   1632
872         .word   -2013126112
873         .word   456
874         .word   771799224
875         .word   696
876         .word   -1828313568
877         .word   1936
878         .word   1426406712
879         .word   1040
880         .word   -66849808
881         .word   568
882         .word   2046945768
883         .word   1376
884         .word   -939318496
885         .word   1848
886         .word   -1174213912
887         .word   344
888         .word   838912200
889         .word   1192
890         .word   -435971176
891         .word   1280
892         .word   -1073544448
893         .word   1216
894         .word   419695624
895         .word   1672
896         .word   -1644004744
897         .word   1016
898         .word   -1559828768
899         .word   816
900         .word   1140920592
901         .word   1008
902         .word   1409372496
903         .word   1368
904         .word   990151808
905         .word   1048
906         .word   184828992
907         .word   1616
908         .word   -1946013136
909         .word   328
910         .word   -955811984
911         .word   1688
912         .word   1795540416
913         .word   480
914         .word   671129760
915         .word   968
916         .word   -1492715792
917         .word   1808
918         .word   -1140657424
```



```
919         .word   232
920         .word   369121368
921         .word   944
922         .word   -1392058664
923         .word   472
924         .word   -620296448
925         .word   688
926         .word   1677824400
927         .word   624
928         .word   1946276304
929         .word   240
930         .word   335564880
931         .word   1752
932         .word   -1845343672
933         .word   80
934         .word   201338928
935         .word   864
936         .word   1208033568
937         .word   1824
938         .word   -1207770400
939         .word   744
940         .word   -1626991088
941         .word   880
942         .word   -1123639656
943         .word   1912
944         .word   1124427104
945         .word   1328
946         .word   -1006431472
947         .word   1344
948         .word   956599432
949         .word   1312
950         .word   822389928
951         .word   440
952         .word   -754505952
953         .word   1112
954         .word   -234632248
955         .word   400
956         .word   -720945352
957         .word   536
958         .word   -1962523072
959         .word   712
960         .word   1845606840
961         .word   1464
962         .word   -637310104
963         .word   1120
964         .word   17067112
965         .word   800
966         .word   -1324962136
967         .word   1680
968         .word   -1677561232
969         .word   1792
970         .word   1225084232
971         .word   1440
972         .word   -670866592
973         .word   2000
974         .word   -1409109328
975         .word   56
976         .word   -217602144
977         .word   296
978         .word   -821602480
979         .word   1400
980         .word   -905762008
981         .word   1136
982         .word   -201075760
983         .word   1864
984         .word   1191540080
```

```
 985         .word   192
 986         .word   268451904
 987         .word   1704
 988         .word   1862653392
 989         .word   1088
 990         .word   -268188736
 991         .word   888
 992         .word   1241590056
 993         .word   912
 994         .word   1543598448
 995         .word   288
 996         .word   939581664
 997         .word   1928
 998         .word   1459959088
 999         .word   1592
1000         .word   1929749920
1001         .word   648
1002         .word   -1761200592
1003         .word   280
1004         .word   -888715456
1005         .word   992
1006         .word   -1593381144
1007         .word   1248
1008         .word   -402414688
1009         .word   264
1010         .word   1040251128
1011         .word   1768
1012         .word   -1778230696
1013         .word   1760
1014         .word   1627778536
1015         .word   1072
1016         .word   218389592
1017         .word   1064
1018         .word   251941968
1019         .word   1152
1020         .word   -536640640
1021         .word   528
1022         .word   2080502256
1023         .word   1568
1024         .word   1896197544
1025         .word   1360
1026         .word   -872205520
1027         .word   1728
1028         .word   -1878900160
1029         .word   40
1030         .word   100669464
1031         .word   8
1032         .word   -150489168
1033         .word   144
1034         .word   469790832
1035         .word   1304
1036         .word   -1039987960
1037         .word   760
1038         .word   1778493864
1039         .word   1992
1040         .word   -1375552840
1041         .word   1664
1042         .word   1761988040
1043         .word   1160
1044         .word   386151472
1045         .word   704
1046         .word   -1727656440
1047         .word   312
1048         .word   973138152
1049         .word   1480
1050         .word   654636272
```



```
1051         .word   448
1052         .word   -653848824
1053         .word   152
1054         .word   -351811648
1055         .word   1432
1056         .word   721732800
1057         .word   408
1058         .word   570460296
1059         .word   1496
1060         .word   -771536056
1061         .word   896
1062         .word   -1459171640
1063         .word   1096
1064         .word   117732464
1065         .word   1336
1066         .word   855942304
1067         .word   1456
1068         .word   755293400
1069         .word   272
1070         .word   1006694640
1071         .word   1168
1072         .word   352599096
1073         .word   256
1074         .word   -922267832
1075         .word   584
1076         .word   -2029619600
1077         .word   2040
1078         .word   -1442665816
1079         .word   960
1080         .word   1342259520
1081         .word   976
1082         .word   -1526268168
1083         .word   1144
1084         .word   50619488
1085         .word   1984
1086         .word   1493503240
1087         .word   1024
1088         .word   151276616
1089         .word   184
1090         .word   436234344
1091         .word   1744
1092         .word   1694891512
1093         .word   392
1094         .word   -687392976
1095         .word   1584
1096         .word   -2080239088
1097         .word   1472
1098         .word   -805092544
1099         .word   1560
1100         .word   -2113795576
1101         .word   1408
1102         .word   688180424
1103         .word   952
1104         .word   1510041960
1105         .word   136
1106         .word   503347320
1107         .word   1624
1108         .word   2063959424
1109         .word   2016
1110         .word   -1476222304
1111         .word   1712
1112         .word   1829101016
1113         .word   464
1114         .word   738242736
1115         .type   Te1,#object
1116         .size   Te1,2048
```

```
1117 !
1118 ! CONSTANT POOL
1119 !
1120 Te2:
1121         .word   792
1122         .word   -1526320360
1123         .word   992
1124         .word   -2079865888
1125         .word   952
1126         .word   -1727564872
1127         .word   984
1128         .word   -1928875048
1129         .word   1936
1130         .word   218627984
1131         .word   856
1132         .word   -1123634344
1133         .word   888
1134         .word   -1324944520
1135         .word   1576
1136         .word   1409584680
1137         .word   384
1138         .word   1342374272
1139         .word   8
1140         .word   50335752
1141         .word   824
1142         .word   -1459195080
1143         .word   344
1144         .word   2097328472
1145         .word   2032
1146         .word   419905520
1147         .word   1720
1148         .word   1644539576
1149         .word   1368
1150         .word   -436048552
1151         .word   944
1152         .word   -1710791760
1153         .word   1616
1154         .word   1157922384
1155         .word   1040
1156         .word   -1660879856
1157         .word   1608
1158         .word   1074024008
1159         .word   1000
1160         .word   -2029530136
1161         .word   2000
1162         .word   352813008
1163         .word   712
1164         .word   -351956280
1165         .word   568
1166         .word   -922455496
1167         .word   1920
1168         .word   185065344
1169         .word   1384
1170         .word   -335409816
1171         .word   1696
1172         .word   1728421536
1173         .word   1296
1174         .word   -50135792
1175         .word   1400
1176         .word   -368956040
1177         .word   1248
1178         .word   -1090446112
1179         .word   1312
1180         .word   -150823648
1181         .word   912
1182         .word   -1777917040
```



```
1183         .word   1536
1184         .word   1527045632
1185         .word   1464
1186         .word   -1039946312
1187         .word   2024
1188         .word   470224872
1189         .word   1176
1190         .word   -1375605608
1191         .word   304
1192         .word   1778540848
1193         .word   432
1194         .word   1510171056
1195         .word   504
1196         .word   1090777592
1197         .word   1976
1198         .word   34058168
1199         .word   1632
1200         .word   1325669984
1201         .word   416
1202         .word   1543717280
1203         .word   1320
1204         .word   -201159384
1205         .word   1832
1206         .word   872845096
1207         .word   1928
1208         .word   134729608
1209         .word   904
1210         .word   -1828252792
1211         .word   1728
1212         .word   1929731776
1213         .word   392
1214         .word   1392710024
1215         .word   168
1216         .word   1057050792
1217         .word   32
1218         .word   201343008
1219         .word   1592
1220         .word   1376038456
1221         .word   280
1222         .word   1694642456
1223         .word   1560
1224         .word   1577381400
1225         .word   192
1226         .word   671187136
1227         .word   1200
1228         .word   -1593721680
1229         .word   40
1230         .word   251678760
1231         .word   1232
1232         .word   -1258193712
1233         .word   56
1234         .word   151023672
1235         .word   144
1236         .word   906043536
1237         .word   1024
1238         .word   -1694442496
1239         .word   1808
1240         .word   1023868688
1241         .word   1880
1242         .word   637955928
1243         .word   312
1244         .word   1761767736
1245         .word   1424
1246         .word   -855376496
1247         .word   936
1248         .word   -1626909784
```

```
1249         .word   72
1250         .word   453021768
1251         .word   1048
1252         .word   -1644106728
1253         .word   352
1254         .word   1946337632
1255         .word   208
1256         .word   771858640
1257         .word   216
1258         .word   755085528
1259         .word   880
1260         .word   -1308171408
1261         .word   720
1262         .word   -301620528
1263         .word   1280
1264         .word   -83698432
1265         .word   656
1266         .word   -167435632
1267         .word   472
1268         .word   1292087768
1269         .word   1712
1270         .word   1627766448
1271         .word   1432
1272         .word   -838603368
1273         .word   328
1274         .word   2063765832
1275         .word   1816
1276         .word   1040641816
1277         .word   376
1278         .word   1896018296
1279         .word   1056
1280         .word   -1761567712
1281         .word   664
1282         .word   -184208744
1283         .word   1672
1284         .word   1745211016
1285         .word   0
1286         .word   0
1287         .word   1896
1288         .word   738594664
1289         .word   256
1290         .word   1610744064
1291         .word   2016
1292         .word   520560608
1293         .word   1416
1294         .word   -939274872
1295         .word   728
1296         .word   -318393640
1297         .word   848
1298         .word   -1106861232
1299         .word   1624
1300         .word   1174695512
1301         .word   1520
1302         .word   -654098960
1303         .word   456
1304         .word   1258525128
1305         .word   592
1306         .word   -570121648
1307         .word   608
1308         .word   -737885600
1309         .word   704
1310         .word   -402292032
1311         .word   1656
1312         .word   1241788024
1313         .word   1664
1314         .word   1795546752
```



```
1315         .word   1912
1316         .word   705048440
1317         .word   1360
1318         .word   -452821680
1319         .word   2008
1320         .word   369586136
1321         .word   536
1322         .word   -989580776
1323         .word   616
1324         .word   -687549848
1325         .word   408
1326         .word   1426272664
1327         .word   1064
1328         .word   -1811903448
1329         .word   552
1330         .word   -821800408
1331         .word   1992
1332         .word   268914632
1333         .word   16
1334         .word   100671504
1335         .word   1016
1336         .word   -2130185224
1337         .word   640
1338         .word   -268107136
1339         .word   480
1340         .word   1141096928
1341         .word   1272
1342         .word   -1174328072
1343         .word   1344
1344         .word   -486384320
1345         .word   648
1346         .word   -217771384
1347         .word   1304
1348         .word   -33362664
1349         .word   512
1350         .word   -1073479168
1351         .word   1144
1352         .word   -1979700104
1353         .word   1168
1354         .word   -1392378736
1355         .word   1256
1356         .word   -1140781848
1357         .word   448
1358         .word   1208189376
1359         .word   1960
1360         .word   67604392
1361         .word   1504
1362         .word   -553443872
1363         .word   1456
1364         .word   -1056719440
1365         .word   1744
1366         .word   1963294416
1367         .word   264
1368         .word   1661079816
1369         .word   128
1370         .word   805372032
1371         .word   2040
1372         .word   436678648
1373         .word   1944
1374         .word   235401112
1375         .word   1680
1376         .word   1829109392
1377         .word   1640
1378         .word   1275334248
1379         .word   96
1380         .word   335593568
```

```
1381         .word   152
1382         .word   889270424
1383         .word   1888
1384         .word   788930400
1385         .word   760
1386         .word   -519703816
1387         .word   1208
1388         .word   -1576948552
1389         .word   544
1390         .word   -872136160
1391         .word   184
1392         .word   956395704
1393         .word   1568
1394         .word   1459920416
1395         .word   1336
1396         .word   -234705608
1397         .word   1008
1398         .word   -2113412112
1399         .word   488
1400         .word   1191432680
1401         .word   800
1402         .word   -1408875744
1403         .word   744
1404         .word   -419048728
1405         .word   200
1406         .word   721522888
1407         .word   920
1408         .word   -1794690152
1409         .word   768
1410         .word   -1610218752
1411         .word   1032
1412         .word   -1744778232
1413         .word   632
1414         .word   -788204936
1415         .word   1760
1416         .word   2131042016
1417         .word   272
1418         .word   1711415568
1419         .word   336
1420         .word   2114101584
1421         .word   1152
1422         .word   -1425941376
1423         .word   1088
1424         .word   -2097128384
1425         .word   560
1426         .word   -905682384
1427         .word   1904
1428         .word   688275312
1429         .word   1472
1430         .word   -754754112
1431         .word   160
1432         .word   1006715040
1433         .word   1776
1434         .word   2030386928
1435         .word   752
1436         .word   -502930704
1437         .word   88
1438         .word   486584408
1439         .word   1752
1440         .word   1980067544
1441         .word   1792
1442         .word   990306048
1443         .word   400
1444         .word   1443045776
1445         .word   464
1446         .word   1308860880
```



```
1447         .word   80
1448         .word   503357520
1449         .word   584
1450         .word   -620457400
1451         .word   48
1452         .word   167796784
1453         .word   288
1454         .word   1812087072
1455         .word   736
1456         .word   -469384480
1457         .word   1552
1458         .word   1560608272
1459         .word   1688
1460         .word   1845882520
1461         .word   1376
1462         .word   -285074080
1463         .word   784
1464         .word   -1509547248
1465         .word   1160
1466         .word   -1476277112
1467         .word   1192
1468         .word   -1543402328
1469         .word   1824
1470         .word   923180832
1471         .word   968
1472         .word   -1962437688
1473         .word   1848
1474         .word   839298872
1475         .word   1600
1476         .word   1124359744
1477         .word   440
1478         .word   1493397944
1479         .word   872
1480         .word   -1224289432
1481         .word   1128
1482         .word   -1946153880
1483         .word   1704
1484         .word   1678085800
1485         .word   624
1486         .word   -771431824
1487         .word   1352
1488         .word   -536720056
1489         .word   864
1490         .word   -1274625184
1491         .word   688
1492         .word   -100310352
1493         .word   1952
1494         .word   117940128
1495         .word   1872
1496         .word   621182800
1497         .word   808
1498         .word   -1358539992
1499         .word   976
1500         .word   -1912101936
1501         .word   1392
1502         .word   -385729168
1503         .word   64
1504         .word   402686016
1505         .word   1488
1506         .word   -721191472
1507         .word   960
1508         .word   -2012773440
1509         .word   296
1510         .word   1862422824
1511         .word   368
1512         .word   1912791408
```

```
1513         .word   224
1514         .word   604094688
1515         .word   1328
1516         .word   -251478736
1517         .word   1440
1518         .word   -956064352
1519         .word   1584
1520         .word   1359265328
1521         .word   1856
1522         .word   587620160
1523         .word   1768
1524         .word   2080706280
1525         .word   928
1526         .word   -1677245536
1527         .word   248
1528         .word   553775352
1529         .word   600
1530         .word   -586894760
1531         .word   1512
1532         .word   -603779608
1533         .word   1112
1534         .word   -2046792616
1535         .word   1104
1536         .word   -2063565744
1537         .word   896
1538         .word   -1878588544
1539         .word   496
1540         .word   1107550704
1541         .word   1448
1542         .word   -1006400088
1543         .word   816
1544         .word   -1442421968
1545         .word   576
1546         .word   -670793152
1547         .word   24
1548         .word   83898392
1549         .word   1968
1550         .word   17285040
1551         .word   112
1552         .word   302047344
1553         .word   776
1554         .word   -1559883000
1555         .word   424
1556         .word   1594053032
1557         .word   696
1558         .word   -117083464
1559         .word   1480
1560         .word   -805089848
1561         .word   1072
1562         .word   -1862222800
1563         .word   1544
1564         .word   1476709896
1565         .word   232
1566         .word   654430440
1567         .word   1264
1568         .word   -1191101200
1569         .word   1800
1570         .word   939970312
1571         .word   1984
1572         .word   319250368
1573         .word   1216
1574         .word   -1291756352
1575         .word   136
1576         .word   855707784
1577         .word   840
1578         .word   -1157196984
```



```
1579         .word   1736
1580         .word   1879396040
1581         .word   1136
1582         .word   -1996473232
1583         .word   1184
1584         .word   -1493066592
1585         .word   1240
1586         .word   -1241420584
1587         .word   240
1588         .word   570548464
1589         .word   1080
1590         .word   -1845449672
1591         .word   1864
1592         .word   537284424
1593         .word   1648
1594         .word   1225014896
1595         .word   680
1596         .word   -16428376
1597         .word   320
1598         .word   2013430080
1599         .word   1784
1600         .word   2047160056
1601         .word   1120
1602         .word   -1895818144
1603         .word   1288
1604         .word   -134034168
1605         .word   1096
1606         .word   -2147464120
1607         .word   104
1608         .word   385929320
1609         .word   1528
1610         .word   -637325832
1611         .word   1840
1612         .word   822525744
1613         .word   528
1614         .word   -972807664
1615         .word   832
1616         .word   -1207532736
1617         .word   520
1618         .word   -1023143416
1619         .word   1224
1620         .word   -1342092088
1621         .word   360
1622         .word   1996673384
1623         .word   120
1624         .word   285274232
1625         .word   1408
1626         .word   -888939136
1627         .word   672
1628         .word   -66764128
1629         .word   1496
1630         .word   -704418344
1631         .word   176
1632         .word   973168816
1633         .type   Te2,#object
1634         .size   Te2,2048
1635 !
1636 ! CONSTANT POOL
1637 !
1638 Te3:
1639         .word   792
1640         .word   1661283888
1641         .word   992
1642         .word   2080647104
1643         .word   952
1644         .word   1996803952
```

```
1645         .word   984
1646         .word   2063888304
1647         .word   1936
1648         .word   -234852360
1649         .word   856
1650         .word   1795550896
1651         .word   888
1652         .word   1862635248
1653         .word   1576
1654         .word   -989682552
1655         .word   384
1656         .word   805470976
1657         .word   8
1658         .word   16783376
1659         .word   824
1660         .word   1728401008
1661         .word   344
1662         .word   721676976
1663         .word   2032
1664         .word   -33501384
1665         .word   1720
1666         .word   -687663704
1667         .word   1368
1668         .word   -1425591704
1669         .word   944
1670         .word   1980028768
1671         .word   1616
1672         .word   -905827208
1673         .word   1040
1674         .word   -2113607432
1675         .word   1608
1676         .word   -922614712
1677         .word   1000
1678         .word   2097430480
1679         .word   2000
1680         .word   -100618376
1681         .word   712
1682         .word   1493654928
1683         .word   568
1684         .word   1191595120
1685         .word   1920
1686         .word   -268410920
1687         .word   1384
1688         .word   -1392025080
1689         .word   1696
1690         .word   -737985128
1691         .word   1296
1692         .word   -1576539400
1693         .word   1400
1694         .word   -1358474712
1695         .word   1248
1696         .word   -1677330152
1697         .word   1312
1698         .word   -1542997352
1699         .word   912
1700         .word   1912911648
1701         .word   1536
1702         .word   -1073554216
1703         .word   1464
1704         .word   -1224338520
1705         .word   2024
1706         .word   -50272504
1707         .word   1176
1708         .word   -1828359704
1709         .word   304
1710         .word   637751904
```



```
1711         .word   432
1712         .word   906154848
1713         .word   504
1714         .word   1057098736
1715         .word   1976
1716         .word   -150988888
1717         .word   1632
1718         .word   -872252392
1719         .word   416
1720         .word   872604480
1721         .word   1320
1722         .word   -1526226296
1723         .word   1832
1724         .word   -452876664
1725         .word   1928
1726         .word   -251639864
1727         .word   904
1728         .word   1896128272
1729         .word   1728
1730         .word   -670851752
1731         .word   392
1732         .word   822254352
1733         .word   168
1734         .word   352450896
1735         .word   32
1736         .word   67133504
1737         .word   1592
1738         .word   -956132184
1739         .word   280
1740         .word   587409968
1741         .word   1560
1742         .word   -1023216408
1743         .word   192
1744         .word   402735488
1745         .word   1200
1746         .word   -1778054728
1747         .word   40
1748         .word   83916880
1749         .word   1232
1750         .word   -1710904968
1751         .word   56
1752         .word   117459056
1753         .word   144
1754         .word   302100768
1755         .word   1024
1756         .word   -2147165992
1757         .word   1808
1758         .word   -503189768
1759         .word   1880
1760         .word   -352242072
1761         .word   312
1762         .word   654527088
1763         .word   1424
1764         .word   -1308201992
1765         .word   936
1766         .word   1963261776
1767         .word   72
1768         .word   151050384
1769         .word   1048
1770         .word   -2096828184
1771         .word   352
1772         .word   738435776
1773         .word   208
1774         .word   436302240
1775         .word   216
1776         .word   453077424
```

```
1777         .word   880
1778         .word   1845860064
1779         .word   720
1780         .word   1510438304
1781         .word   1280
1782         .word   -1610097960
1783         .word   656
1784         .word   1376236832
1785         .word   472
1786         .word   990014384
1787         .word   1712
1788         .word   -704442952
1789         .word   1432
1790         .word   -1291422744
1791         .word   328
1792         .word   688118416
1793         .word   1816
1794         .word   -486410520
1795         .word   376
1796         .word   788761328
1797         .word   1056
1798         .word   -2080065384
1799         .word   664
1800         .word   1393012016
1801         .word   1672
1802         .word   -788314680
1803         .word   0
1804         .word   0
1805         .word   1896
1806         .word   -318675448
1807         .word   256
1808         .word   537068032
1809         .word   2016
1810         .word   -67043560
1811         .word   1416
1812         .word   -1324989496
1813         .word   728
1814         .word   1527213488
1815         .word   848
1816         .word   1778775712
1817         .word   1624
1818         .word   -889047960
1819         .word   1520
1820         .word   -1106851016
1821         .word   456
1822         .word   956455824
1823         .word   592
1824         .word   1241969824
1825         .word   608
1826         .word   1275503808
1827         .word   704
1828         .word   1476871552
1829         .word   1656
1830         .word   -821930968
1831         .word   1664
1832         .word   -805085736
1833         .word   1912
1834         .word   -285125080
1835         .word   1360
1836         .word   -1442370952
1837         .word   2008
1838         .word   -83839128
1839         .word   536
1840         .word   1124478000
1841         .word   616
1842         .word   1292287184
```



```
1843         .word   408
1844         .word   855812912
1845         .word   1064
1846         .word   -2063294328
1847         .word   552
1848         .word   1158052944
1849         .word   1992
1850         .word   -117405880
1851         .word   16
1852         .word   33566752
1853         .word   1016
1854         .word   2130972656
1855         .word   640
1856         .word   1342670080
1857         .word   480
1858         .word   1006773184
1859         .word   1272
1860         .word   -1627008728
1861         .word   1344
1862         .word   -1475929512
1863         .word   648
1864         .word   1359453456
1865         .word   1304
1866         .word   -1559760152
1867         .word   512
1868         .word   1074136064
1869         .word   1144
1870         .word   -1895542744
1871         .word   1168
1872         .word   -1845138952
1873         .word   1256
1874         .word   -1660559096
1875         .word   448
1876         .word   939672448
1877         .word   1960
1878         .word   -184539256
1879         .word   1504
1880         .word   -1140393192
1881         .word   1456
1882         .word   -1241117768
1883         .word   1744
1884         .word   -637293192
1885         .word   264
1886         .word   553851408
1887         .word   128
1888         .word   268534016
1889         .word   2040
1890         .word   -16722136
1891         .word   1944
1892         .word   -218073112
1893         .word   1680
1894         .word   -771527176
1895         .word   1640
1896         .word   -855481336
1897         .word   96
1898         .word   201367744
1899         .word   152
1900         .word   318875952
1901         .word   1888
1902         .word   -335446504
1903         .word   760
1904         .word   1594297840
1905         .word   1208
1906         .word   -1761275480
1907         .word   544
1908         .word   1141269568
```

```
1909         .word   184
1910         .word   385993072
1911         .word   1568
1912         .word   -1006453608
1913         .word   1336
1914         .word   -1492675928
1915         .word   1008
1916         .word   2114197472
1917         .word   488
1918         .word   1023556560
1919         .word   800
1920         .word   1678075456
1921         .word   744
1922         .word   1560755664
1923         .word   200
1924         .word   419518864
1925         .word   920
1926         .word   1929686832
1927         .word   768
1928         .word   1610941952
1929         .word   1032
1930         .word   -2130394936
1931         .word   632
1932         .word   1325829360
1933         .word   1760
1934         .word   -603718376
1935         .word   272
1936         .word   570634784
1937         .word   336
1938         .word   704901792
1939         .word   1152
1940         .word   -1878697512
1941         .word   1088
1942         .word   -2012997544
1943         .word   560
1944         .word   1174819936
1945         .word   1904
1946         .word   -301904328
1947         .word   1472
1948         .word   -1207526568
1949         .word   160
1950         .word   335667520
1951         .word   1776
1952         .word   -570176200
1953         .word   752
1954         .word   1577522656
1955         .word   88
1956         .word   184608944
1957         .word   1752
1958         .word   -620513944
1959         .word   1792
1960         .word   -536748328
1961         .word   400
1962         .word   839037728
1963         .word   464
1964         .word   973239200
1965         .word   80
1966         .word   167833760
1967         .word   584
1968         .word   1225186448
1969         .word   48
1970         .word   100683872
1971         .word   288
1972         .word   604201536
1973         .word   736
1974         .word   1543972288
```



```
1975         .word   1552
1976         .word   -1039995656
1977         .word   1688
1978         .word   -754747928
1979         .word   1376
1980         .word   -1408796136
1981         .word   784
1982         .word   1644508704
1983         .word   1160
1984         .word   -1861926456
1985         .word   1192
1986         .word   -1794825848
1987         .word   1824
1988         .word   -469647720
1989         .word   968
1990         .word   2030329744
1991         .word   1848
1992         .word   -419326296
1993         .word   1600
1994         .word   -939385768
1995         .word   440
1996         .word   922930032
1997         .word   872
1998         .word   1829093072
1999         .word   1128
2000         .word   -1929093112
2001         .word   1704
2002         .word   -721214072
2003         .word   624
2004         .word   1309054176
2005         .word   1352
2006         .word   -1459158456
2007         .word   864
2008         .word   1812309696
2009         .word   688
2010         .word   1443353952
2011         .word   1952
2012         .word   -201310312
2013         .word   1872
2014         .word   -369021320
2015         .word   808
2016         .word   1694858832
2017         .word   976
2018         .word   2047113120
2019         .word   1392
2020         .word   -1375253960
2021         .word   64
2022         .word   134267008
2023         .word   1488
2024         .word   -1173968008
2025         .word   960
2026         .word   2013546368
2027         .word   296
2028         .word   620984912
2029         .word   368
2030         .word   771986144
2031         .word   224
2032         .word   469836224
2033         .word   1328
2034         .word   -1509455176
2035         .word   1440
2036         .word   -1274659944
2037         .word   1584
2038         .word   -972911432
2039         .word   1856
2040         .word   -402579880
```

```
2041         .word   1768
2042         .word   -586947320
2043         .word   928
2044         .word   1946478400
2045         .word   248
2046         .word   520161776
2047         .word   600
2048         .word   1258745008
2049         .word   1512
2050         .word   -1123622136
2051         .word   1112
2052         .word   -1962659736
2053         .word   1104
2054         .word   -1979438984
2055         .word   896
2056         .word   1879344896
2057         .word   496
2058         .word   1040323552
2059         .word   1448
2060         .word   -1257888888
2061         .word   816
2062         .word   1711625824
2063         .word   576
2064         .word   1208403072
2065         .word   24
2066         .word   50341936
2067         .word   1968
2068         .word   -167768136
2069         .word   112
2070         .word   234918112
2071         .word   776
2072         .word   1627725328
2073         .word   424
2074         .word   889387856
2075         .word   696
2076         .word   1460129136
2077         .word   1480
2078         .word   -1190755512
2079         .word   1072
2080         .word   -2046523208
2081         .word   1544
2082         .word   -1056783160
2083         .word   232
2084         .word   486619600
2085         .word   1264
2086         .word   -1643787976
2087         .word   1800
2088         .word   -519977272
2089         .word   1984
2090         .word   -134176936
2091         .word   1216
2092         .word   -1744463528
2093         .word   136
2094         .word   285317392
2095         .word   840
2096         .word   1761992336
2097         .word   1736
2098         .word   -654080696
2099         .word   1136
2100         .word   -1912321992
2101         .word   1184
2102         .word   -1811596904
2103         .word   1240
2104         .word   -1694125720
2105         .word   240
2106         .word   503386592
```



```
2107         .word   1080
2108         .word   -2029743960
2109         .word   1864
2110         .word   -385808824
2111         .word   1648
2112         .word   -838710216
2113         .word   680
2114         .word   1426586960
2115         .word   320
2116         .word   671335040
2117         .word   1784
2118         .word   -553396952
2119         .word   1120
2120         .word   -1945864168
2121         .word   1288
2122         .word   -1593326904
2123         .word   1096
2124         .word   -1996226488
2125         .word   104
2126         .word   218151120
2127         .word   1528
2128         .word   -1090071768
2129         .word   1840
2130         .word   -436105544
2131         .word   528
2132         .word   1107702816
2133         .word   832
2134         .word   1745208960
2135         .word   520
2136         .word   1090919440
2137         .word   1224
2138         .word   -1727692472
2139         .word   360
2140         .word   755219152
2141         .word   120
2142         .word   251693296
2143         .word   1408
2144         .word   -1341760552
2145         .word   672
2146         .word   1409803584
2147         .word   1496
2148         .word   -1157188760
2149         .word   176
2150         .word   369217888
2151         .type   Te3,#object
2152         .size   Te3,2048
2153         .align  4
2154 !
2155 ! CONSTANT POOL
2156 !
2157 Te4:
2158         .word   1667457891
2159         .word   2088533116
2160         .word   2004318071
2161         .word   2071690107
2162         .word   -218959118
2163         .word   1802201963
2164         .word   1869573999
2165         .word   -976894523
2166         .word   808464432
2167         .word   16843009
2168         .word   1734829927
2169         .word   724249387
2170         .word   -16843010
2171         .word   -673720361
2172         .word   -1414812757
```

```
2173         .word   1987475062
2174         .word   -892679478
2175         .word   -2105376126
2176         .word   -909522487
2177         .word   2105376125
2178         .word   -84215046
2179         .word   1499027801
2180         .word   1195853639
2181         .word   -252645136
2182         .word   -1381126739
2183         .word   -724249388
2184         .word   -1566399838
2185         .word   -1347440721
2186         .word   -1667457892
2187         .word   -1532713820
2188         .word   1920103026
2189         .word   -1061109568
2190         .word   -1212696649
2191         .word   -33686019
2192         .word   -1819044973
2193         .word   640034342
2194         .word   909522486
2195         .word   1061109567
2196         .word   -134744073
2197         .word   -858993460
2198         .word   875836468
2199         .word   -1515870811
2200         .word   -437918235
2201         .word   -235802127
2202         .word   1903260017
2203         .word   -656877352
2204         .word   825307441
2205         .word   353703189
2206         .word   67372036
2207         .word   -943208505
2208         .word   589505315
2209         .word   -1010580541
2210         .word   404232216
2211         .word   -1768515946
2212         .word   84215045
2213         .word   -1701143910
2214         .word   117901063
2215         .word   303174162
2216         .word   -2139062144
2217         .word   -488447262
2218         .word   -336860181
2219         .word   656877351
2220         .word   -1296911694
2221         .word   1970632053
2222         .word   151587081
2223         .word   -2088533117
2224         .word   741092396
2225         .word   437918234
2226         .word   454761243
2227         .word   1852730990
2228         .word   1515870810
2229         .word   -1600085856
2230         .word   1381126738
2231         .word   993737531
2232         .word   -690563370
2233         .word   -1280068685
2234         .word   690563369
2235         .word   -471604253
2236         .word   791621423
2237         .word   -2071690108
2238         .word   1397969747
```



```
2239         .word   -774778415
2240         .word   0
2241         .word   -303174163
2242         .word   538976288
2243         .word   -50529028
2244         .word   -1313754703
2245         .word   1532713819
2246         .word   1785358954
2247         .word   -875836469
2248         .word   -1094795586
2249         .word   960051513
2250         .word   1246382666
2251         .word   1280068684
2252         .word   1482184792
2253         .word   -808464433
2254         .word   -791621424
2255         .word   -269488145
2256         .word   -1431655766
2257         .word   -67372037
2258         .word   1128481603
2259         .word   1296911693
2260         .word   858993459
2261         .word   -2054847099
2262         .word   1162167621
2263         .word   -101058055
2264         .word   33686018
2265         .word   2139062143
2266         .word   1347440720
2267         .word   1010580540
2268         .word   -1616928865
2269         .word   -1465341784
2270         .word   1364283729
2271         .word   -1549556829
2272         .word   1077952576
2273         .word   -1886417009
2274         .word   -1835887982
2275         .word   -1650614883
2276         .word   943208504
2277         .word   -168430091
2278         .word   -1128481604
2279         .word   -1229539658
2280         .word   -623191334
2281         .word   555819297
2282         .word   269488144
2283         .word   -1
2284         .word   -202116109
2285         .word   -757935406
2286         .word   -842150451
2287         .word   202116108
2288         .word   320017171
2289         .word   -320017172
2290         .word   1600085855
2291         .word   -1751672937
2292         .word   1145324612
2293         .word   387389207
2294         .word   -993737532
2295         .word   -1482184793
2296         .word   2122219134
2297         .word   1027423549
2298         .word   1684300900
2299         .word   1566399837
2300         .word   421075225
2301         .word   1936946035
2302         .word   1616928864
2303         .word   -2122219135
2304         .word   1330597711
```

```
2305         .word   -589505316
2306         .word   572662306
2307         .word   707406378
2308         .word   -1869574000
2309         .word   -2004318072
2310         .word   1179010630
2311         .word   -286331154
2312         .word   -1195853640
2313         .word   336860180
2314         .word   -555819298
2315         .word   1583242846
2316         .word   185273099
2317         .word   -606348325
2318         .word   -522133280
2319         .word   842150450
2320         .word   976894522
2321         .word   168430090
2322         .word   1229539657
2323         .word   101058054
2324         .word   606348324
2325         .word   1549556828
2326         .word   -1027423550
2327         .word   -741092397
2328         .word   -1397969748
2329         .word   1650614882
2330         .word   -1852730991
2331         .word   -1785358955
2332         .word   -454761244
2333         .word   2038004089
2334         .word   -404232217
2335         .word   -926365496
2336         .word   926365495
2337         .word   1835887981
2338         .word   -1920103027
2339         .word   -707406379
2340         .word   1313754702
2341         .word   -1448498775
2342         .word   1819044972
2343         .word   1448498774
2344         .word   -185273100
2345         .word   -353703190
2346         .word   1701143909
2347         .word   2054847098
2348         .word   -1364283730
2349         .word   134744072
2350         .word   -1162167622
2351         .word   2021161080
2352         .word   623191333
2353         .word   774778414
2354         .word   471604252
2355         .word   -1499027802
2356         .word   -1263225676
2357         .word   -960051514
2358         .word   -387389208
2359         .word   -572662307
2360         .word   1953789044
2361         .word   522133279
2362         .word   1263225675
2363         .word   -1111638595
2364         .word   -1953789045
2365         .word   -1970632054
2366         .word   1886417008
2367         .word   1044266558
2368         .word   -1246382667
2369         .word   1717986918
2370         .word   1212696648
```



```
2371         .word   50529027
2372         .word   -151587082
2373         .word   235802126
2374         .word   1633771873
2375         .word   892679477
2376         .word   1465341783
2377         .word   -1179010631
2378         .word   -2038004090
2379         .word   -1044266559
2380         .word   488447261
2381         .word   -1633771874
2382         .word   -505290271
2383         .word   -117901064
2384         .word   -1734829928
2385         .word   286331153
2386         .word   1768515945
2387         .word   -640034343
2388         .word   -1903260018
2389         .word   -1802201964
2390         .word   -1684300901
2391         .word   505290270
2392         .word   -2021161081
2393         .word   -370546199
2394         .word   -825307442
2395         .word   1431655765
2396         .word   673720360
2397         .word   -538976289
2398         .word   -1936946036
2399         .word   -1583242847
2400         .word   -1987475063
2401         .word   218959117
2402         .word   -1077952577
2403         .word   -421075226
2404         .word   1111638594
2405         .word   1751672936
2406         .word   1094795585
2407         .word   -1717986919
2408         .word   757935405
2409         .word   252645135
2410         .word   -1330597712
2411         .word   1414812756
2412         .word   -1145324613
2413         .word   370546198
2414         .type   Te4,#object
2415         .size   Te4,1024
2416         .align  8


2419         .section        ".data",#alloc,#write
2420         .align  8
2421 aes_const:

2423 #ifdef __sparcv9
2424 !
2425 ! for v8plus, the addresses are 64-bit long so we should use .xword
2426 ! instead of .word
2427 !
2428 !
2429         .xword  (Te0+0x0)
2430         .xword  (Te1+0x0)
2431         .xword  (Te2+0x0)
2432         .xword  (Te3+0x0)
2433         .xword  (Te4+0x0)
2434         .xword  (Td0+0x0)
2435         .xword  (Td1+0x0)
2436         .xword  (Td2+0x0)
```

```
2437         .xword  (Td3+0x0)
2438         .xword  (Td4+0x0)

2440 #else /* __sparcv9 */

2442 !
2443 ! for v8plus, the addresses are 32-bit long, we use filler 0's so that
2444 ! we can use ldx to load the addresses just like in the v9 version
2445 !
2446         .word   0
2447         .word   (Te0+0x0)
2448         .word   0
2449         .word   (Te1+0x0)
2450         .word   0
2451         .word   (Te2+0x0)
2452         .word   0
2453         .word   (Te3+0x0)
2454         .word   0
2455         .word   (Te4+0x0)
2456         .word   0
2457         .word   (Td0+0x0)
2458         .word   0
2459         .word   (Td1+0x0)
2460         .word   0
2461         .word   (Td2+0x0)
2462         .word   0
2463         .word   (Td3+0x0)
2464         .word   0
2465         .word   (Td4+0x0)

2467 #endif /* __sparcv9 */

2469         .type   aes_const,#object
2470         .size   aes_const,40

2472         .section        ".text",#alloc,#execinstr
2473 !
2474 ! SUBROUTINE aes_encrypt_impl
2475 !
2476 ! void aes_encrypt_impl(const uint32_t rk[], int Nr, const uint32_t pt[4],
2477 !    uint32_t ct[4]);
2478 !
2479 ! OFFSET    SOURCE LINE LABEL   INSTRUCTION

2481         .global aes_encrypt_impl
2482         aes_encrypt_impl:

2484 #ifdef __sparcv9

2486         save    %sp, -SA(MINFRAME), %sp

2488 #ifdef PIC
2489         sethi   %hi(aes_const), %l0
2490 .L1:
2491         call    .+8

2493         sethi   %hi(_GLOBAL_OFFSET_TABLE_-(.L1-.)), %l1
2494         or      %l0, %lo(aes_const), %l0
2495         ld      [%i2], %g1

2497         or      %l1, %lo(_GLOBAL_OFFSET_TABLE_-(.L1-.)), %l1
2498         sethi   %hi(0xff000000), %i5

2500         add     %l1, %o7, %l1
2501         sethi   %hi(0x7fff8), %l6
```



```
2503         and     %g1, %i5, %o0
2504         or      %l6, %lo(0x7ff8), %l6

2506         sll     %g1, 8, %o5
2507         ldx     [%l1+%l0], %l7
2508 #else /* PIC */
2509         sethi   %hh(aes_const), %l0
2510         sethi   %lm(aes_const), %l7

2512         or      %l0, %hm(aes_const), %l0
2513         or      %l7, %lo(aes_const), %l7

2515         sllx    %l0, 32, %l0
2516         sethi   %hi(0xff000000), %i5
2517         ld      [%i2], %g1

2519         sethi   %hi(0x7fff8), %l6
2520         or      %l0, %l7, %l7

2522         and     %g1, %i5, %o0
2523         or      %l6, %lo(0x7ff8), %l6

2525         sll     %g1, 8, %o5
2526 #endif /* PIC */

2528 #else /* __sparcv9 */

2530         save    %sp, -SA(MINFRAME), %sp
2531         sethi   %hi(aes_const), %l0

2533 #ifdef PIC
2534 .L1:
2535         call    .+8

2537         sethi   %hi(_GLOBAL_OFFSET_TABLE_-(.L1-.)), %l1
2538         or      %l0, %lo(aes_const), %l0
2539         ld      [%i2], %g1

2541         or      %l1, %lo(_GLOBAL_OFFSET_TABLE_-(.L1-.)), %l1
2542         sethi   %hi(0xff000000), %i5

2544         add     %l1, %o7, %l1
2545         sethi   %hi(0x7fff8), %l6

2547         and     %g1, %i5, %o0
2548         or      %l6, %lo(0x7ff8), %l6

2550         sll     %g1, 8, %o5
2551         ld      [%l1+%l0], %l7
2552 #else /* PIC */
2553         or      %l0, %lo(aes_const), %l7
2554         sethi   %hi(0xff000000), %i5
2555         ld      [%i2], %g1

2557         sethi   %hi(0x7fff8), %l6

2559         and     %g1, %i5, %o0
2560         or      %l6, %lo(0x7ff8), %l6

2562         sll     %g1, 8, %o5
2563 #endif /* PIC */

2565 #endif /* __sparcv9 */

2567         sll     %g1, 3, %g1
2568         and     %o5, %i5, %o5
```

```
2569         ld      [%i2 + 4], %g2

2571         sllx    %o0, 11, %o0
2572         and     %g1, %l6, %g1

2574         or      %o0, %o5, %o5
2575         ldx     [%i0], %o1
2576         and     %g2, %i5, %o0

2578         sll     %g2, 8, %g5
2579         or      %o5, %g1, %g1
2580         ldx     [%l7], %l0

2582         sll     %g2, 3, %g2
2583         xor     %g1, %o1, %o1
2584         ldx     [%l7 + 8], %l1

2586         sllx    %o0, 11, %o0
2587         and     %g5, %i5, %o5
2588         ld      [%i2 + 8], %g3

2590         or      %o0, %o5, %o5
2591         and     %g2, %l6, %g2
2592         ldx     [%i0 + 8], %o2

2594         or      %o5, %g2, %g2
2595         and     %g3, %i5, %o5
2596         ldx     [%l7 + 16], %l2

2598         sll     %g3, 8, %o0
2599         xor     %g2, %o2, %o2
2600         ldx     [%l7 + 24], %l3

2602         sllx    %o5, 11, %o5
2603         and     %o0, %i5, %o0
2604         ld      [%i2 + 12], %g4

2606         sll     %g3, 3, %g1
2607         or      %o5, %o0, %o0
2608         ldx     [%l7 + 32], %i4

2610         and     %g1, %l6, %g1
2611         and     %g4, %i5, %o5
2612         ldx     [%i0 + 16], %o3

2614         sll     %g4, 8, %g3
2615         or      %o0, %g1, %g1

2617         sll     %g4, 3, %g5
2618         xor     %g1, %o3, %o3

2620         sllx    %o5, 11, %g4
2621         and     %g3, %i5, %o0
2622         ldx     [%i0 + 24], %o5

2624         srlx    %o1, 32, %l4
2625         or      %g4, %o0, %o0

2627         srl     %o2, 21, %l5
2628         and     %g5, %l6, %g5

2630 .L2:
2631         srl     %o3, 8, %l6
2632         xor     %g5, %o0, %g5
2633         ldx     [%l0 + %l4], %o0
```



```
2635         and     %l6, 0x7f8, %l6
2636         xor     %g5, %o5, %o4
2637         ldx     [%l1 + %l5], %o5

2639         and     %o4, 0x7f8, %l7
2640         add     %i0, 32, %i0
2641         ldx     [%l2 + %l6], %g5

2643         srlx    %o2, 32, %l4
2644         xor     %o0, %o5, %o5
2645         ldx     [%l3 + %l7], %o0

2647         srl     %o3, 21, %l5
2648         xor     %o5, %g5, %g5
2649         ldx     [%i0], %o5

2651         srl     %o4, 8, %l6
2652         xor     %g5, %o0, %g5
2653         ldx     [%l0 + %l4], %o0

2655         and     %l6, 0x7f8, %l6
2656         xor     %g5, %o5, %g1
2657         ldx     [%l1 + %l5], %o5

2659         and     %o1, 0x7f8, %l7
2660         ldx     [%l2 + %l6], %g5

2662         srlx    %o3, 32, %l4
2663         xor     %o0, %o5, %o5
2664         ldx     [%l3 + %l7], %o0

2666         srl     %o4, 21, %l5
2667         xor     %o5, %g5, %g5
2668         ldx     [%i0 + 8], %o5

2670         srl     %o1, 8, %l6
2671         xor     %g5, %o0, %g5
2672         ldx     [%l0 + %l4], %o0

2674         and     %l6, 0x7f8, %l6
2675         xor     %g5, %o5, %g2
2676         ldx     [%l1 + %l5], %o5

2678         and     %o2, 0x7f8, %l7
2679         ldx     [%l2 + %l6], %g5

2681         srlx    %o4, 32, %l4
2682         xor     %o0, %o5, %o5
2683         ldx     [%l3 + %l7], %o0

2685         srl     %o1, 21, %l5
2686         xor     %o5, %g5, %g5
2687         ldx     [%i0 + 16], %o5

2689         srl     %o2, 8, %l6
2690         xor     %g5, %o0, %g5
2691         ldx     [%l0 + %l4], %o0

2693         and     %l6, 0x7f8, %l6
2694         xor     %g5, %o5, %g3
2695         ldx     [%l1 + %l5], %o5

2697         and     %o3, 0x7f8, %l7
2698         ldx     [%l2 + %l6], %g5
2699         sub     %i1, 2, %i1
```

```
2701         srlx    %g1, 32, %l4
2702         xor     %o0, %o5, %o5
2703         ldx     [%l3 + %l7], %o0

2705         srl     %g2, 21, %l5
2706         xor     %o5, %g5,%g5
2707         ldx     [%i0 + 24], %o5

2709 .L3:
2710         srl     %g3, 8, %l6
2711         xor     %g5, %o0, %g5
2712         ldx     [%l0 + %l4], %o0

2714         and     %l6, 0x7f8, %l6
2715         xor     %g5, %o5, %g4
2716         ldx     [%l1 + %l5], %o5

2718         and     %g4, 0x7f8, %l7
2719         add     %i0, 32, %i0
2720         ldx     [%l2 + %l6], %g5

2722         srlx    %g2, 32, %l4
2723         xor     %o0, %o5, %o5
2724         ldx     [%l3 + %l7], %o0

2726         srl     %g3, 21, %l5
2727         xor     %o5, %g5,%g5
2728         ldx     [%i0], %o5

2730         srl     %g4, 8, %l6
2731         xor     %g5, %o0, %g5
2732         ldx     [%l0 + %l4], %o0

2734         and     %l6, 0x7f8, %l6
2735         xor     %g5, %o5, %o1
2736         ldx     [%l1 + %l5], %o5

2738         and     %g1, 0x7f8, %l7
2739         ldx     [%l2 + %l6], %g5

2741         srlx    %g3, 32, %l4
2742         xor     %o0, %o5, %o5
2743         ldx     [%l3 + %l7], %o0

2745         srl     %g4, 21, %l5
2746         xor     %o5, %g5,%g5
2747         ldx     [%i0 + 8], %o5

2749         srl     %g1, 8, %l6
2750         xor     %g5, %o0, %g5
2751         ldx     [%l0 + %l4], %o0

2753         and     %l6, 0x7f8, %l6
2754         xor     %g5, %o5, %o2
2755         ldx     [%l1 + %l5], %o5

2757         and     %g2, 0x7f8, %l7
2758         ldx     [%l2 + %l6], %g5

2760         srlx    %g4, 32, %l4
2761         xor     %o0, %o5, %o5
2762         ldx     [%l3 + %l7], %o0

2764         srl     %g1, 21, %l5
2765         xor     %o5, %g5,%g5
2766         ldx     [%i0 + 16], %o5
```



```
2768         srl     %g2, 8, %l6
2769         xor     %g5, %o0, %g5
2770         ldx     [%l0 + %l4], %o0

2772         and     %l6, 0x7f8, %l6
2773         xor     %g5, %o5, %o3
2774         ldx     [%l1 + %l5], %o5

2776         and     %g3, 0x7f8, %l7
2777         ldx     [%l2 + %l6], %g5

2779         srlx    %o1, 32, %l4
2780         xor     %o0, %o5, %o5
2781         ldx     [%l3 + %l7], %o0

2783         srl     %o2, 21, %l5
2784         xor     %o5, %g5,%g5
2785         ldx     [%i0 + 24], %o5


2788         srl     %o3, 8, %l6
2789         xor     %g5, %o0, %g5
2790         ldx     [%l0 + %l4], %o0

2792         and     %l6, 0x7f8, %l6
2793         xor     %g5, %o5, %o4
2794         ldx     [%l1 + %l5], %o5

2796         and     %o4, 0x7f8, %l7
2797         add     %i0, 32, %i0
2798         ldx     [%l2 + %l6], %g5

2800         srlx    %o2, 32, %l4
2801         xor     %o0, %o5, %o5
2802         ldx     [%l3 + %l7], %o0

2804         srl     %o3, 21, %l5
2805         xor     %o5, %g5, %g5
2806         ldx     [%i0], %o5

2808         srl     %o4, 8, %l6
2809         xor     %g5, %o0, %g5
2810         ldx     [%l0 + %l4], %o0

2812         and     %l6, 0x7f8, %l6
2813         xor     %g5, %o5, %g1
2814         ldx     [%l1 + %l5], %o5

2816         and     %o1, 0x7f8, %l7
2817         ldx     [%l2 + %l6], %g5

2819         srlx    %o3, 32, %l4
2820         xor     %o0, %o5, %o5
2821         ldx     [%l3 + %l7], %o0

2823         srl     %o4, 21, %l5
2824         xor     %o5, %g5, %g5
2825         ldx     [%i0 + 8], %o5

2827         srl     %o1, 8, %l6
2828         xor     %g5, %o0, %g5
2829         ldx     [%l0 + %l4], %o0

2831         and     %l6, 0x7f8, %l6
2832         xor     %g5, %o5, %g2
```

```
2833         ldx     [%l1 + %l5], %o5

2835         and     %o2, 0x7f8, %l7
2836         ldx     [%l2 + %l6], %g5

2838         srlx    %o4, 32, %l4
2839         xor     %o0, %o5, %o5
2840         ldx     [%l3 + %l7], %o0

2842         srl     %o1, 21, %l5
2843         xor     %o5, %g5, %g5
2844         ldx     [%i0 + 16], %o5

2846         srl     %o2, 8, %l6
2847         xor     %g5, %o0, %g5
2848         ldx     [%l0 + %l4], %o0

2850         and     %l6, 0x7f8, %l6
2851         xor     %g5, %o5, %g3
2852         ldx     [%l1 + %l5], %o5

2854         and     %o3, 0x7f8, %l7
2855         ldx     [%l2 + %l6], %g5
2856         subcc   %i1, 2, %i1

2858         srlx    %g1, 32, %l4
2859         xor     %o0, %o5, %o5
2860         ldx     [%l3 + %l7], %o0

2862         srl     %g2, 21, %l5
2863         xor     %o5, %g5, %g5
2864         bnz,pt  %icc, .L3
2865         ldx     [%i0 + 24], %o5


2868 .L4:
2869         srl     %l4, 1, %l4     !***** should be removed after
2870         srl     %l5, 1, %l5     !***** unrolling the loop and correcting
2871         add     %i0, 32, %i0    !***** the last iteration

2873         srl     %i5, 8, %i1
2874         xor     %g5, %o0, %g5
2875         ld      [%i4 + %l4], %o1

2877         srl     %g3, 9, %l6
2878         xor     %g5, %o5, %g4
2879         ld      [%i4 + %l5], %o2

2881         srl     %g4, 1, %l7
2882         and     %l6, 0x3fc, %l6

2884         srl     %i1, 8, %l0
2885         and     %l7, 0x3fc, %l7
2886         ld      [%i4 + %l6], %o3

2888         and     %o1, %i5, %o1
2889         and     %o2, %i1, %o2
2890         ld      [%i4 + %l7], %o4

2892         srlx    %g2, 33, %l4
2893         xor     %o1, %o2, %o2
2894         ld      [%i0], %o5

2896         srl     %g3, 22, %l5
2897         and     %o3, %l0, %o3
2898         ld      [%i4 + %l4], %l1
```



```
2900         xor     %o2, %o3, %o3
2901         and     %o4, 0xff, %o4

2903         srl     %g4, 9, %l6
2904         xor     %o3, %o4, %o4
2905         ld      [%i4 + %l5], %l2

2907         srl     %g1, 1, %l7
2908         and     %l6, 0x3fc, %l6

2910         xor     %o4, %o5, %o5
2911         and     %l7, 0x3fc, %l7
2912         ld      [%i4 + %l6], %l3

2914         and     %l1, %i5, %l1
2915         and     %l2, %i1, %l2
2916         st      %o5, [%i3]

2918         srlx    %g3, 33, %l6
2919         xor     %l1, %l2, %l2
2920         ld      [%i4 + %l7], %l4

2922         srl     %g4, 22, %l5
2923         and     %l3, %l0, %l3
2924         ld      [%i0 + 4], %g5

2926         xor     %l2, %l3, %l3
2927         and     %l4, 0xff, %l4
2928         ld      [%i4 + %l6], %o1

2930         srl     %g1, 9, %l6
2931         xor     %l3, %l4, %l4
2932         ld      [%i4 + %l5], %o2

2934         srl     %g2, 1, %l7
2935         and     %l6, 0x3fc, %l6

2937         xor     %l4, %g5, %g5
2938         and     %l7, 0x3fc, %l7
2939         ld      [%i4 + %l6], %o3

2941         and     %o1, %i5, %o1
2942         and     %o2, %i1, %o2
2943         st      %g5, [%i3 + 4]

2945         srlx    %g4, 33, %l4
2946         xor     %o1, %o2, %o2
2947         ld      [%i4 + %l7], %o4

2949         srl     %g1, 22, %l5
2950         and     %o3, %l0, %o3
2951         ld      [%i0 + 8], %o5

2953         xor     %o2, %o3, %o3
2954         and     %o4, 0xff, %o4
2955         ld      [%i4 + %l4], %l1

2957         srl     %g2, 9, %l6
2958         xor     %o3, %o4, %o4
2959         ld      [%i4 + %l5], %l2

2961         srl     %g3, 1, %l7
2962         and     %l6, 0x3fc, %l6

2964         xor     %o4, %o5, %o5
```

```
2965         and     %l7, 0x3fc, %l7
2966         ld      [%i4 + %l6], %l3

2968         and     %l1, %i5, %l1
2969         and     %l2, %i1, %l2
2970         ld      [%i4 + %l7], %l4

2972         xor     %l1, %l2, %l2
2973         and     %l3, %l0, %l3
2974         ld      [%i0 + 12], %l5

2976         xor     %l2, %l3, %l3
2977         and     %l4, 0xff, %l4
2978         st      %o5, [%i3 + 8]

2980         xor     %l3, %l4, %l4

2982         xor     %l4, %l5, %l5
2983         st      %l5, [%i3 + 12]

2985         ret
2986         restore %g0,%g0,%g0

2988         .type   aes_encrypt_impl,2
2989         .size   aes_encrypt_impl,(.-aes_encrypt_impl)


2992         .section        ".text",#alloc,#execinstr
2993         .align  8192
2994 !
2995 ! SUBROUTINE aes_decrypt_impl
2996 !
2997 ! void aes_decrypt_impl(const uint32_t rk[], int Nr, const uint32_t ct[4],
2998 !    uint32_t pt[4]);
2999 !
3000 ! OFFSET    SOURCE LINE LABEL   INSTRUCTION

3002         .global aes_decrypt_impl
3003         aes_decrypt_impl:

3005 #ifdef __sparcv9

3007         save    %sp, -SA(MINFRAME), %sp

3009 #ifdef PIC
3010         sethi   %hi(aes_const), %l0
3011 .L1d:
3012         call    .+8

3014         sethi   %hi(_GLOBAL_OFFSET_TABLE_-(.L1d-.)), %l1
3015         or      %l0, %lo(aes_const), %l0
3016         ld      [%i2 + 12], %g1

3018         or      %l1, %lo(_GLOBAL_OFFSET_TABLE_-(.L1d-.)), %l1
3019         sethi   %hi(0xff000000), %i5

3021         add     %l1, %o7, %l1
3022         sethi   %hi(0x7fff8), %l6

3024         and     %g1, %i5, %o0
3025         or      %l6, %lo(0x7ff8), %l6

3027         sll     %g1, 8, %o5
3028         ldx     [%l1+%l0], %l7
3029 #else /* PIC */
3030         sethi   %hh(aes_const), %l0
```



```
3031         sethi   %lm(aes_const), %l7

3033         or      %l0, %hm(aes_const), %l0
3034         or      %l7, %lo(aes_const), %l7

3036         sllx    %l0, 32, %l0
3037         sethi   %hi(0xff000000), %i5
3038         ld      [%i2 + 12], %g1

3040         sethi   %hi(0x7fff8), %l6
3041         or      %l0, %l7, %l7

3043         and     %g1, %i5, %o0
3044         or      %l6, %lo(0x7ff8), %l6

3046         sll     %g1, 8, %o5
3047 #endif /* PIC */

3049 #else /* __sparcv9 */

3051         save    %sp, -SA(MINFRAME), %sp
3052         sethi   %hi(aes_const), %l0

3054 #ifdef PIC
3055 .L1d:
3056         call    .+8

3058         sethi   %hi(_GLOBAL_OFFSET_TABLE_-(.L1d-.)), %l1
3059         or      %l0, %lo(aes_const), %l0
3060         ld      [%i2 + 12], %g1

3062         or      %l1, %lo(_GLOBAL_OFFSET_TABLE_-(.L1d-.)), %l1
3063         sethi   %hi(0xff000000), %i5

3065         add     %l1, %o7, %l1
3066         sethi   %hi(0x7fff8), %l6

3068         and     %g1, %i5, %o0
3069         or      %l6, %lo(0x7ff8), %l6

3071         sll     %g1, 8, %o5
3072         ld      [%l1+%l0], %l7
3073 #else /* PIC */
3074         or      %l0, %lo(aes_const), %l7
3075         sethi   %hi(0xff000000), %i5
3076         ld      [%i2 + 12], %g1

3078         sethi   %hi(0x7fff8), %l6

3080         and     %g1, %i5, %o0
3081         or      %l6, %lo(0x7ff8), %l6

3083         sll     %g1, 8, %o5
3084 #endif /* PIC */

3086 #endif /* __sparcv9 */

3088         sll     %g1, 3, %g1
3089         and     %o5, %i5, %o5
3090         ld      [%i2 + 8], %g2

3092         sllx    %o0, 11, %o0
3093         and     %g1, %l6, %g1

3095         or      %o0, %o5, %o5
3096         ldx     [%i0], %o1
```

```
3097         and     %g2, %i5, %o0

3099         sll     %g2, 8, %g5
3100         or      %o5, %g1, %g1
3101         ldx     [%l7 + 40], %l0

3103         sll     %g2, 3, %g2
3104         xor     %g1, %o1, %o1
3105         ldx     [%l7 + 48], %l1

3107         sllx    %o0, 11, %o0
3108         and     %g5, %i5, %o5
3109         ld      [%i2 + 4], %g3

3111         or      %o0, %o5, %o5
3112         and     %g2, %l6, %g2
3113         ldx     [%i0 + 8], %o2

3115         or      %o5, %g2, %g2
3116         and     %g3, %i5, %o5
3117         ldx     [%l7 + 56], %l2

3119         sll     %g3, 8, %o0
3120         xor     %g2, %o2, %o2
3121         ldx     [%l7 + 64], %l3

3123         sllx    %o5, 11, %o5
3124         and     %o0, %i5, %o0
3125         ld      [%i2], %g4

3127         sll     %g3, 3, %g1
3128         or      %o5, %o0, %o0
3129         ldx     [%l7 + 72], %i4

3131         and     %g1, %l6, %g1
3132         and     %g4, %i5, %o5
3133         ldx     [%i0 + 16], %o3

3135         sll     %g4, 8, %g3
3136         or      %o0, %g1, %g1

3138         sll     %g4, 3, %g5
3139         xor     %g1, %o3, %o3

3141         sllx    %o5, 11, %g4
3142         and     %g3, %i5, %o0
3143         ldx     [%i0 + 24], %o5

3145         srlx    %o1, 32, %l4
3146         or      %g4, %o0, %o0

3148         srl     %o2, 21, %l5
3149         and     %g5, %l6, %g5

3151 .L2d:
3152         srl     %o3, 8, %l6
3153         xor     %g5, %o0, %g5
3154         ldx     [%l0 + %l4], %o0

3156         and     %l6, 0x7f8, %l6
3157         xor     %g5, %o5, %o4
3158         ldx     [%l1 + %l5], %o5

3160         and     %o4, 0x7f8, %l7
3161         add     %i0, 32, %i0
3162         ldx     [%l2 + %l6], %g5
```



```
3164         srlx    %o2, 32, %l4
3165         xor     %o0, %o5, %o5
3166         ldx     [%l3 + %l7], %o0

3168         srl     %o3, 21, %l5
3169         xor     %o5, %g5, %g5
3170         ldx     [%i0], %o5

3172         srl     %o4, 8, %l6
3173         xor     %g5, %o0, %g5
3174         ldx     [%l0 + %l4], %o0

3176         and     %l6, 0x7f8, %l6
3177         xor     %g5, %o5, %g1
3178         ldx     [%l1 + %l5], %o5

3180         and     %o1, 0x7f8, %l7
3181         ldx     [%l2 + %l6], %g5

3183         srlx    %o3, 32, %l4
3184         xor     %o0, %o5, %o5
3185         ldx     [%l3 + %l7], %o0

3187         srl     %o4, 21, %l5
3188         xor     %o5, %g5, %g5
3189         ldx     [%i0 + 8], %o5

3191         srl     %o1, 8, %l6
3192         xor     %g5, %o0, %g5
3193         ldx     [%l0 + %l4], %o0

3195         and     %l6, 0x7f8, %l6
3196         xor     %g5, %o5, %g2
3197         ldx     [%l1 + %l5], %o5

3199         and     %o2, 0x7f8, %l7
3200         ldx     [%l2 + %l6], %g5

3202         srlx    %o4, 32, %l4
3203         xor     %o0, %o5, %o5
3204         ldx     [%l3 + %l7], %o0

3206         srl     %o1, 21, %l5
3207         xor     %o5, %g5, %g5
3208         ldx     [%i0 + 16], %o5

3210         srl     %o2, 8, %l6
3211         xor     %g5, %o0, %g5
3212         ldx     [%l0 + %l4], %o0

3214         and     %l6, 0x7f8, %l6
3215         xor     %g5, %o5, %g3
3216         ldx     [%l1 + %l5], %o5

3218         and     %o3, 0x7f8, %l7
3219         ldx     [%l2 + %l6], %g5
3220         sub     %i1, 2, %i1

3222         srlx    %g1, 32, %l4
3223         xor     %o0, %o5, %o5
3224         ldx     [%l3 + %l7], %o0

3226         srl     %g2, 21, %l5
3227         xor     %o5, %g5,%g5
3228         ldx     [%i0 + 24], %o5
```

```
3230 .L3d:
3231         srl     %g3, 8, %l6
3232         xor     %g5, %o0, %g5
3233         ldx     [%l0 + %l4], %o0

3235         and     %l6, 0x7f8, %l6
3236         xor     %g5, %o5, %g4
3237         ldx     [%l1 + %l5], %o5

3239         and     %g4, 0x7f8, %l7
3240         add     %i0, 32, %i0
3241         ldx     [%l2 + %l6], %g5

3243         srlx    %g2, 32, %l4
3244         xor     %o0, %o5, %o5
3245         ldx     [%l3 + %l7], %o0

3247         srl     %g3, 21, %l5
3248         xor     %o5, %g5,%g5
3249         ldx     [%i0], %o5

3251         srl     %g4, 8, %l6
3252         xor     %g5, %o0, %g5
3253         ldx     [%l0 + %l4], %o0

3255         and     %l6, 0x7f8, %l6
3256         xor     %g5, %o5, %o1
3257         ldx     [%l1 + %l5], %o5

3259         and     %g1, 0x7f8, %l7
3260         ldx     [%l2 + %l6], %g5

3262         srlx    %g3, 32, %l4
3263         xor     %o0, %o5, %o5
3264         ldx     [%l3 + %l7], %o0

3266         srl     %g4, 21, %l5
3267         xor     %o5, %g5,%g5
3268         ldx     [%i0 + 8], %o5

3270         srl     %g1, 8, %l6
3271         xor     %g5, %o0, %g5
3272         ldx     [%l0 + %l4], %o0

3274         and     %l6, 0x7f8, %l6
3275         xor     %g5, %o5, %o2
3276         ldx     [%l1 + %l5], %o5

3278         and     %g2, 0x7f8, %l7
3279         ldx     [%l2 + %l6], %g5

3281         srlx    %g4, 32, %l4
3282         xor     %o0, %o5, %o5
3283         ldx     [%l3 + %l7], %o0

3285         srl     %g1, 21, %l5
3286         xor     %o5, %g5,%g5
3287         ldx     [%i0 + 16], %o5

3289         srl     %g2, 8, %l6
3290         xor     %g5, %o0, %g5
3291         ldx     [%l0 + %l4], %o0

3293         and     %l6, 0x7f8, %l6
3294         xor     %g5, %o5, %o3
```



```
3295         ldx     [%l1 + %l5], %o5

3297         and     %g3, 0x7f8, %l7
3298         ldx     [%l2 + %l6], %g5

3300         srlx    %o1, 32, %l4
3301         xor     %o0, %o5, %o5
3302         ldx     [%l3 + %l7], %o0

3304         srl     %o2, 21, %l5
3305         xor     %o5, %g5,%g5
3306         ldx     [%i0 + 24], %o5


3309         srl     %o3, 8, %l6
3310         xor     %g5, %o0, %g5
3311         ldx     [%l0 + %l4], %o0

3313         and     %l6, 0x7f8, %l6
3314         xor     %g5, %o5, %o4
3315         ldx     [%l1 + %l5], %o5

3317         and     %o4, 0x7f8, %l7
3318         add     %i0, 32, %i0
3319         ldx     [%l2 + %l6], %g5

3321         srlx    %o2, 32, %l4
3322         xor     %o0, %o5, %o5
3323         ldx     [%l3 + %l7], %o0

3325         srl     %o3, 21, %l5
3326         xor     %o5, %g5, %g5
3327         ldx     [%i0], %o5

3329         srl     %o4, 8, %l6
3330         xor     %g5, %o0, %g5
3331         ldx     [%l0 + %l4], %o0

3333         and     %l6, 0x7f8, %l6
3334         xor     %g5, %o5, %g1
3335         ldx     [%l1 + %l5], %o5

3337         and     %o1, 0x7f8, %l7
3338         ldx     [%l2 + %l6], %g5

3340         srlx    %o3, 32, %l4
3341         xor     %o0, %o5, %o5
3342         ldx     [%l3 + %l7], %o0

3344         srl     %o4, 21, %l5
3345         xor     %o5, %g5, %g5
3346         ldx     [%i0 + 8], %o5

3348         srl     %o1, 8, %l6
3349         xor     %g5, %o0, %g5
3350         ldx     [%l0 + %l4], %o0

3352         and     %l6, 0x7f8, %l6
3353         xor     %g5, %o5, %g2
3354         ldx     [%l1 + %l5], %o5

3356         and     %o2, 0x7f8, %l7
3357         ldx     [%l2 + %l6], %g5

3359         srlx    %o4, 32, %l4
3360         xor     %o0, %o5, %o5
```

```
3361         ldx     [%l3 + %l7], %o0

3363         srl     %o1, 21, %l5
3364         xor     %o5, %g5, %g5
3365         ldx     [%i0 + 16], %o5

3367         srl     %o2, 8, %l6
3368         xor     %g5, %o0, %g5
3369         ldx     [%l0 + %l4], %o0

3371         and     %l6, 0x7f8, %l6
3372         xor     %g5, %o5, %g3
3373         ldx     [%l1 + %l5], %o5

3375         and     %o3, 0x7f8, %l7
3376         ldx     [%l2 + %l6], %g5
3377         subcc   %i1, 2, %i1

3379         srlx    %g1, 32, %l4
3380         xor     %o0, %o5, %o5
3381         ldx     [%l3 + %l7], %o0

3383         srl     %g2, 21, %l5
3384         xor     %o5, %g5, %g5
3385         bnz,pt  %icc, .L3d
3386         ldx     [%i0 + 24], %o5


3389 .L4d:
3390         srl     %l4, 1, %l4     !***** should be removed after
3391         srl     %l5, 1, %l5     !***** unrolling the loop and correcting
3392         add     %i0, 32, %i0    !***** the last iteration

3394         srl     %i5, 8, %i1
3395         xor     %g5, %o0, %g5
3396         ld      [%i4 + %l4], %o1

3398         srl     %g3, 9, %l6
3399         xor     %g5, %o5, %g4
3400         ld      [%i4 + %l5], %o2

3402         srl     %g4, 1, %l7
3403         and     %l6, 0x3fc, %l6

3405         srl     %i1, 8, %l0
3406         and     %l7, 0x3fc, %l7
3407         ld      [%i4 + %l6], %o3

3409         and     %o1, %i5, %o1
3410         and     %o2, %i1, %o2
3411         ld      [%i4 + %l7], %o4

3413         srlx    %g2, 33, %l4
3414         xor     %o1, %o2, %o2
3415         ld      [%i0], %o5

3417         srl     %g3, 22, %l5
3418         and     %o3, %l0, %o3
3419         ld      [%i4 + %l4], %l1

3421         xor     %o2, %o3, %o3
3422         and     %o4, 0xff, %o4

3424         srl     %g4, 9, %l6
3425         xor     %o3, %o4, %o4
3426         ld      [%i4 + %l5], %l2
```



```
3428         srl     %g1, 1, %l7
3429         and     %l6, 0x3fc, %l6

3431         xor     %o4, %o5, %o5
3432         and     %l7, 0x3fc, %l7
3433         ld      [%i4 + %l6], %l3

3435         and     %l1, %i5, %l1
3436         and     %l2, %i1, %l2
3437         st      %o5, [%i3 + 12]

3439         srlx    %g3, 33, %l6
3440         xor     %l1, %l2, %l2
3441         ld      [%i4 + %l7], %l4

3443         srl     %g4, 22, %l5
3444         and     %l3, %l0, %l3
3445         ld      [%i0 + 4], %g5

3447         xor     %l2, %l3, %l3
3448         and     %l4, 0xff, %l4
3449         ld      [%i4 + %l6], %o1

3451         srl     %g1, 9, %l6
3452         xor     %l3, %l4, %l4
3453         ld      [%i4 + %l5], %o2

3455         srl     %g2, 1, %l7
3456         and     %l6, 0x3fc, %l6

3458         xor     %l4, %g5, %g5
3459         and     %l7, 0x3fc, %l7
3460         ld      [%i4 + %l6], %o3

3462         and     %o1, %i5, %o1
3463         and     %o2, %i1, %o2
3464         st      %g5, [%i3 + 8]

3466         srlx    %g4, 33, %l4
3467         xor     %o1, %o2, %o2
3468         ld      [%i4 + %l7], %o4

3470         srl     %g1, 22, %l5
3471         and     %o3, %l0, %o3
3472         ld      [%i0 + 8], %o5

3474         xor     %o2, %o3, %o3
3475         and     %o4, 0xff, %o4
3476         ld      [%i4 + %l4], %l1

3478         srl     %g2, 9, %l6
3479         xor     %o3, %o4, %o4
3480         ld      [%i4 + %l5], %l2

3482         srl     %g3, 1, %l7
3483         and     %l6, 0x3fc, %l6

3485         xor     %o4, %o5, %o5
3486         and     %l7, 0x3fc, %l7
3487         ld      [%i4 + %l6], %l3

3489         and     %l1, %i5, %l1
3490         and     %l2, %i1, %l2
3491         ld      [%i4 + %l7], %l4
```

```
3493         xor     %l1, %l2, %l2
3494         and     %l3, %l0, %l3
3495         ld      [%i0 + 12], %l5

3497         xor     %l2, %l3, %l3
3498         and     %l4, 0xff, %l4
3499         st      %o5, [%i3 + 4]

3501         xor     %l3, %l4, %l4

3503         xor     %l4, %l5, %l5
3504         st      %l5, [%i3]

3506         ret
3507         restore %g0,%g0,%g0

3509         .type   aes_decrypt_impl,2
3510         .size   aes_decrypt_impl,(.-aes_decrypt_impl)


3513         .section        ".text",#alloc
3514         .align  8
3515 !
3516 ! CONSTANT POOL
3517 !
3518 Td0:
3519         .word   648
3520         .word   -200983936
3521         .word   1008
3522         .word   1090726552
3523         .word   208
3524         .word   386213400
3525         .word   464
3526         .word   654505136
3527         .word   472
3528         .word   -1425842600
3529         .word   248
3530         .word   -1660801144
3531         .word   1376
3532         .word   -100481704
3533         .word   600
3534         .word   -486531944
3535         .word   256
3536         .word   805819048
3537         .word   1384
3538         .word   1979936688
3539         .word   1088
3540         .word   -872172408
3541         .word   1960
3542         .word   33710376
3543         .word   632
3544         .word   -452542496
3545         .word   1576
3546         .word   705060536
3547         .word   304
3548         .word   889332736
3549         .word   1448
3550         .word   1644502136
3551         .word   1776
3552         .word   -1325215160
3553         .word   296
3554         .word   -1174349000
3555         .word   552
3556         .word   -369068864
3557         .word   744
3558         .word   -33159416
```



```
3559         .word   1560
3560         .word   788768784
3561         .word   1032
3562         .word   1275560080
3563         .word   1128
3564         .word   1174715672
3565         .word   856
3566         .word   -754463184
3567         .word   24
3568         .word   -1895629000
3569         .word   168
3570         .word   -1845173080
3571         .word   1528
3572         .word   1828968280
3573         .word   1192
3574         .word   1375915728
3575         .word   1696
3576         .word   -1107027608
3577         .word   704
3578         .word   1946226328
3579         .word   584
3580         .word   -536655544
3581         .word   1136
3582         .word   -922336736
3583         .word   936
3584         .word   -1039905968
3585         .word   1952
3586         .word   -1912353856
3587         .word   1224
3588         .word   1476522840
3589         .word   312
3590         .word   -1190949144
3591         .word   1520
3592         .word   -519930448
3593         .word   1920
3594         .word   -2012911432
3595         .word   1608
3596         .word   537223984
3597         .word   1000
3598         .word   -838740576
3599         .word   792
3600         .word   -553496384
3601         .word   1832
3602         .word   436309008
3603         .word   1208
3604         .word   1359059712
3605         .word   784
3606         .word   1392769576
3607         .word   1416
3608         .word   1677967104
3609         .word   1496
3610         .word   1795519520
3611         .word   2032
3612         .word   -2130378528
3613         .word   1992
3614         .word   134306976
3615         .word   896
3616         .word   1208173248
3617         .word   1144
3618         .word   1158146248
3619         .word   1184
3620         .word   -570203080
3621         .word   656
3622         .word   2064106936
3623         .word   1368
3624         .word   1929812248
```

```
3625         .word   912
3626         .word   1258297104
3627         .word   1816
3628         .word   520387256
3629         .word   816
3630         .word   1426413904
3631         .word   1424
3632         .word   -352239560
3633         .word   376
3634         .word   -1257893864
3635         .word   1072
3636         .word   -989602608
3637         .word   1688
3638         .word   922764584
3639         .word   384
3640         .word   671367056
3641         .word   280
3642         .word   -1090179696
3643         .word   16
3644         .word   50550224
3645         .word   1896
3646         .word   369365728
3647         .word   1104
3648         .word   -822025896
3649         .word   1336
3650         .word   2030412944
3651         .word   1944
3652         .word   117938048
3653         .word   624
3654         .word   1762071816
3655         .word   808
3656         .word   -637032856
3657         .word   48
3658         .word   84276904
3659         .word   1672
3660         .word   872616184
3661         .word   1568
3662         .word   -1509428144
3663         .word   416
3664         .word   771923176
3665         .word   1296
3666         .word   -217928448
3667         .word   40
3668         .word   -1979250288
3669         .word   1312
3670         .word   -167289944
3671         .word   88
3672         .word   -2096668216
3673         .word   512
3674         .word   1611103568
3675         .word   752
3676         .word   1896151088
3677         .word   1512
3678         .word   1845527176
3679         .word   496
3680         .word   553932744
3681         .word   1200
3682         .word   -587189784
3683         .word   1768
3684         .word   1040199024
3685         .word   616
3686         .word   -435819984
3687         .word   1160
3688         .word   1409576360
3689         .word   904
3690         .word   -1006442456
```



```
3691         .word   32
3692         .word   101098360
3693         .word   768
3694         .word   1342222328
3695         .word   200
3696         .word   -1744316128
3697         .word   1712
3698         .word   -1123595080
3699         .word   1096
3700         .word   1073880672
3701         .word   824
3702         .word   -653986888
3703         .word   1408
3704         .word   -402516504
3705         .word   56
3706         .word   -1996202944
3707         .word   1848
3708         .word   419617216
3709         .word   968
3710         .word   -939034920
3711         .word   1288
3712         .word   2080395832
3713         .word   992
3714         .word   1107328840
3715         .word   1984
3716         .word   -2080311736
3717         .word   0
3718         .word   0
3719         .word   72
3720         .word   -2147208168
3721         .word   400
3722         .word   721906240
3723         .word   240
3724         .word   285443424
3725         .word   864
3726         .word   1510183536
3727         .word   2024
3728         .word   235405272
3729         .word   120
3730         .word   -2063482192
3731         .word   488
3732         .word   -1375295248
3733         .word   432
3734         .word   755091768
3735         .word   80
3736         .word   252103456
3737         .word   832
3738         .word   1543844104
3739         .word   1240
3740         .word   1526900360
3741         .word   288
3742         .word   906064336
3743         .word   96
3744         .word   167984520
3745         .word   1176
3746         .word   1460091000
3747         .word   1440
3748         .word   -301681008
3749         .word   216
3750         .word   -1694200592
3751         .word   1024
3752         .word   -1073337736
3753         .word   776
3754         .word   -603912944
3755         .word   720
3756         .word   1996643144
```

```
3757         .word   224
3758         .word   302043312
3759         .word   1808
3760         .word   -1828335536
3761         .word   1536
3762         .word   -1610524888
3763         .word   480
3764         .word   570884632
3765         .word   144
3766         .word   453032168
3767         .word   112
3768         .word   151021656
3769         .word   1936
3770         .word   -1962525336
3771         .word   360
3772         .word   -1241168440
3773         .word   160
3774         .word   503664192
3775         .word   696
3776         .word   -251605976
3777         .word   1400
3778         .word   1962949216
3779         .word   1904
3780         .word   -1727599144
3781         .word   1304
3782         .word   2130905064
3783         .word   1976
3784         .word   16856312
3785         .word   736
3786         .word   1913105888
3787         .word   544
3788         .word   1711398440
3789         .word   728
3790         .word   -83627616
3791         .word   1112
3792         .word   1124158384
3793         .word   1624
3794         .word   587609824
3795         .word   1456
3796         .word   -318250176
3797         .word   1472
3798         .word   -469267688
3799         .word   1720
3800         .word   822535760
3801         .word   528
3802         .word   1661216896
3803         .word   152
3804         .word   -1761537536
3805         .word   1056
3806         .word   -973043456
3807         .word   1064
3808         .word   1241588712
3809         .word   1680
3810         .word   -1157500992
3811         .word   1392
3812         .word   -117337976
3813         .word   1592
3814         .word   688196456
3815         .word   232
3816         .word   -1644070312
3817         .word   1760
3818         .word   -1308522600
3819         .word   104
3820         .word   -2046650528
3821         .word   952
3822         .word   -1056498048
```



```
3823         .word   344
3824         .word   -1291799712
3825         .word   1352
3826         .word   1879428296
3827         .word   136
3828         .word   -1811789872
3829         .word   568
3830         .word   -385670896
3831         .word   1344
3832         .word   -66820576
3833         .word   1280
3834         .word   -268306224
3835         .word   688
3836         .word   2097243840
3837         .word   272
3838         .word   855934840
3839         .word   1080
3840         .word   1224898104
3841         .word   1736
3842         .word   939953672
3843         .word   1120
3844         .word   -905635856
3845         .word   1216
3846         .word   -738174544
3847         .word   1328
3848         .word   -184283528
3849         .word   1320
3850         .word   2047275328
3851         .word   1744
3852         .word   -1224445648
3853         .word   504
3854         .word   -1392116448
3855         .word   352
3856         .word   973401888
3857         .word   640
3858         .word   2013565032
3859         .word   848
3860         .word   1594254552
3861         .word   672
3862         .word   2114073360
3863         .word   1968
3864         .word   -1929339376
3865         .word   1152
3866         .word   -670709952
3867         .word   368
3868         .word   956807920
3869         .word   1040
3870         .word   -1023049816
3871         .word   1272
3872         .word   1560544752
3873         .word   840
3874         .word   -805004320
3875         .word   888
3876         .word   -721326776
3877         .word   1656
3878         .word   620795288
3879         .word   1600
3880         .word   -1408972328
3881         .word   128
3882         .word   402910520
3883         .word   1856
3884         .word   -1677517968
3885         .word   1752
3886         .word   990239704
3887         .word   1640
3888         .word   637780040
```

```
3889          .word   880
3890          .word   1493223328
3891          .word   1888
3892          .word   -1710901240
3893          .word   1048
3894          .word   1325716800
3895          .word   1840
3896          .word   -1794936024
3897          .word   1360
3898          .word   -16305168
3899          .word   264
3900          .word   -1140426688
3901          .word   1912
3902          .word   352798512
3903          .word   1488
3904          .word   -419111224
3905          .word   592
3906          .word   1862383216
3907          .word   1872
3908          .word   -1627369824
3909          .word   328
3910          .word   -1341921616
3911          .word   392
3912          .word   -1543137928
3913          .word   336
3914          .word   1057036680
3915          .word   1584
3916          .word   -1526423168
3917          .word   424
3918          .word   -1576847872
3919          .word   928
3920          .word   1309008312
3921          .word   2016
3922          .word   -2113514192
3923          .word   1792
3924          .word   -1878620800
3925          .word   408
3926          .word   -1492729688
3927          .word   1928
3928          .word   67420752
3929          .word   520
3930          .word   -335095880
3931          .word   1016
3932          .word   -855474064
3933          .word   184
3934          .word   -1861766792
3935          .word   944
3936          .word   1292285032
3937          .word   536
3938          .word   -284851608
3939          .word   1632
3940          .word   -1442682208
3941          .word   1824
3942          .word   -1778374920
3943          .word   1264
3944          .word   -788156648
3945          .word   608
3946          .word   1778663640
3947          .word   1544
3948          .word   738262464
3949          .word   560
3950          .word   1694665720
3951          .word   1256
3952          .word   1577537568
3953          .word   8
3954          .word   -1946047768
```



```
3955          .word   2000
3956          .word   -2029804648
3957          .word   2008
3958          .word   184682864
3959          .word   1432
3960          .word   1728113360
3961          .word   1168
3962          .word   -620326256
3963          .word   1864
3964          .word   268611992
3965          .word   872
3966          .word   -704497512
3967          .word   1232
3968          .word   -687666080
3969          .word   440
3970          .word   -1593809968
3971          .word   712
3972          .word   -134175632
3973          .word   1880
3974          .word   318891080
3975          .word   1648
3976          .word   -1459536016
3977          .word   1464
3978          .word   1627802024
3979          .word   1800
3980          .word   470232936
3981          .word   976
3982          .word   1191545312
3983          .word   1248
3984          .word   -771294520
3985          .word   680
3986          .word   -234645000
3987          .word   192
3988          .word   335967176
3989          .word   920
3990          .word   -956187144
3991          .word   664
3992          .word   -150573232
3993          .word   760
3994          .word   -49982760
3995          .word   1784
3996          .word   1023637664
3997          .word   960
3998          .word   1141300272
3999          .word   1616
4000          .word   -1358455800
4001          .word   1480
4002          .word   1745232368
4003          .word   448
4004          .word   604086624
4005          .word   1552
4006          .word   -1560149256
4007          .word   176
4008          .word   486939536
4009          .word   1504
4010          .word   -503240608
4011          .word   320
4012          .word   1006783576
4013          .word   2040
4014          .word   218409480
4015          .word   456
4016          .word   -1476392056
4017          .word   64
4018          .word   201694960
4019          .word   1728
4020          .word   -1274600224
```

```
4021         .word   800
4022         .word   1443236992
4023         .word   984
4024         .word   -888921336
4025         .word   1704
4026         .word   839234432
4027         .word   576
4028         .word   1812128672
4029         .word   1664
4030         .word   -1207780848
4031         .type   Td0,#object
4032         .size   Td0,2048
4033 !
4034 ! CONSTANT POOL
4035 !
4036 Td1:
4037         .word   640
4038         .word   1359455544
4039         .word   664
4040         .word   2114063144
4041         .word   1560
4042         .word   436256032
4043         .word   1200
4044         .word   973159152
4045         .word   1624
4046         .word   990206808
4047         .word   1928
4048         .word   520415784
4049         .word   1368
4050         .word   -1408773440
4051         .word   1176
4052         .word   1258756120
4053         .word   680
4054         .word   536971216
4055         .word   1968
4056         .word   -1392266392
4057         .word   1160
4058         .word   -2012847184
4059         .word   296
4060         .word   -184544672
4061         .word   2016
4062         .word   1325870776
4063         .word   1720
4064         .word   -989768104
4065         .word   1024
4066         .word   637643296
4067         .word   1144
4068         .word   -1258089192
4069         .word   584
4070         .word   -570062128
4071         .word   824
4072         .word   621138136
4073         .word   1216
4074         .word   1158107248
4075         .word   1800
4076         .word   1560802816
4077         .word   16
4078         .word   -1023312984
4079         .word   144
4080         .word   -2130548864
4081         .word   1304
4082         .word   -1929235272
4083         .word   1584
4084         .word   1795596232
4085         .word   1848
4086         .word   50625272
```



```
4087         .word   1192
4088         .word   352621792
4089         .word   1880
4090         .word   -1090294832
4091         .word   1744
4092         .word   -1794993464
4093         .word   360
4094         .word   -737807336
4095         .word   1688
4096         .word   1476632840
4097         .word   328
4098         .word   1225196360
4099         .word   544
4100         .word   -1912189376
4101         .word   848
4102         .word   1963332680
4103         .word   960
4104         .word   -201034808
4105         .word   856
4106         .word   -1727872528
4107         .word   1768
4108         .word   654691208
4109         .word   1456
4110         .word   -1106834824
4111         .word   184
4112         .word   -268155544
4113         .word   816
4114         .word   -922679968
4115         .word   1440
4116         .word   2097574352
4117         .word   192
4118         .word   1661401680
4119         .word   1040
4120         .word   -452931192
4121         .word   768
4122         .word   -1761441384
4123         .word   552
4124         .word   1644338168
4125         .word   1792
4126         .word   -1325194312
4127         .word   1056
4128         .word   -1157407376
4129         .word   224
4130         .word   -33288960
4131         .word   1184
4132         .word   -117423784
4133         .word   704
4134         .word   1879196480
4135         .word   200
4136         .word   -1895682072
4137         .word   1080
4138         .word   -1811483808
4139         .word   1464
4140         .word   1375985600
4141         .word   280
4142         .word   -1425826152
4143         .word   1808
4144         .word   1912756240
4145         .word   696
4146         .word   -486474632
4147         .word   336
4148         .word   1711451480
4149         .word   56
4150         .word   -1308141248
4151         .word   24
4152         .word   788901392
```

```
4153         .word   1232
4154         .word   -2046415912
4155         .word   1320
4156         .word   -754862016
4157         .word   1936
4158         .word   805389368
4159         .word   1424
4160         .word   587595048
4161         .word   1488
4162         .word   33561424
4163         .word   736
4164         .word   -318721008
4165         .word   344
4166         .word   -1979287328
4167         .word   1168
4168         .word   -1492922976
4169         .word   1920
4170         .word   -218087536
4171         .word   1288
4172         .word   1308839696
4173         .word   1640
4174         .word   1694947232
4175         .word   1704
4176         .word   100675056
4177         .word   248
4178         .word   -788421872
4179         .word   1104
4180         .word   -1006290960
4181         .word   1256
4182         .word   872510104
4183         .word   1280
4184         .word   -1576559960
4185         .word   400
4186         .word   84170504
4187         .word   936
4188         .word   -1542998184
4189         .word   456
4190         .word   184819552
4191         .word   1360
4192         .word   1073940344
4193         .word   48
4194         .word   1577291000
4195         .word   648
4196         .word   -1123848064
4197         .word   1992
4198         .word   1040256080
4199         .word   488
4200         .word   -1777932240
4201         .word   1392
4202         .word   -587075544
4203         .word   560
4204         .word   1292318184
4205         .word   1448
4206         .word   -1862097816
4207         .word   40
4208         .word   1896227560
4209         .word   888
4210         .word   67122848
4211         .word   2040
4212         .word   1610776744
4213         .word   288
4214         .word   419743704
4215         .word   1208
4216         .word   -704254136
4217         .word   1632
4218         .word   -1996357096
```



```
4219         .word   952
4220         .word   1728498928
4221         .word   1512
4222         .word   -1341701616
4223         .word   1088
4224         .word   117722200
4225         .word   448
4226         .word   -419378472
4227         .word   1752
4228         .word   2030454640
4229         .word   568
4230         .word   -1593581488
4231         .word   1864
4232         .word   2080510072
4233         .word   1608
4234         .word   -133947152
4235         .word   0
4236         .word   0
4237         .word   1048
4238         .word   151258160
4239         .word   576
4240         .word   838950760
4241         .word   1376
4242         .word   503352192
4243         .word   624
4244         .word   1812124560
4245         .word   2008
4246         .word   -50300936
4247         .word   688
4248         .word   251931072
4249         .word   240
4250         .word   1023768232
4251         .word   312
4252         .word   906062280
4253         .word   800
4254         .word   167804616
4255         .word   264
4256         .word   1745020208
4257         .word   1672
4258         .word   -1694311776
4259         .word   464
4260         .word   604090736
4261         .word   1416
4262         .word   201347896
4263         .word   120
4264         .word   -1828536520
4265         .word   1680
4266         .word   -1274579792
4267         .word   1264
4268         .word   453303432
4269         .word   632
4270         .word   -2147088856
4271         .word   1296
4272         .word   1627840768
4273         .word   840
4274         .word   1510193752
4275         .word   176
4276         .word   469799120
4277         .word   80
4278         .word   -503013936
4279         .word   1832
4280         .word   -1073413808
4281         .word   536
4282         .word   1006704384
4283         .word   232
4284         .word   302045368
```

```
4285         .word   88
4286         .word   234899560
4287         .word   1384
4288         .word   -234594760
4289         .word   1480
4290         .word   755348800
4291         .word   1600
4292         .word   335607112
4293         .word   1064
4294         .word   1460111560
4295         .word   608
4296         .word   -1358714824
4297         .word   1496
4298         .word   -301674776
4299         .word   2024
4300         .word   -1560020224
4301         .word   1272
4302         .word   -150992592
4303         .word   1504
4304         .word   1543739304
4305         .word   1576
4306         .word   1141060056
4307         .word   416
4308         .word   1527241712
4309         .word   944
4310         .word   -1962796728
4311         .word   1760
4312         .word   -889119184
4313         .word   832
4314         .word   -1241026592
4315         .word   792
4316         .word   -1207490680
4317         .word   1616
4318         .word   -687763744
4319         .word   128
4320         .word   1107500072
4321         .word   512
4322         .word   319076624
4323         .word   256
4324         .word   -2079969144
4325         .word   1000
4326         .word   -2063445728
4327         .word   1984
4328         .word   -771368472
4329         .word   136
4330         .word   -1375221360
4331         .word   872
4332         .word   -956216056
4333         .word   600
4334         .word   486863224
4335         .word   1944
4336         .word   -603614848
4337         .word   1888
4338         .word   218378896
4339         .word   1664
4340         .word   1996885784
4341         .word   864
4342         .word   721787056
4343         .word   1224
4344         .word   -1459386936
4345         .word   2000
4346         .word   285516352
4347         .word   272
4348         .word   1191660320
4349         .word   1568
4350         .word   -1475877792
```



```
4351         .word   208
4352         .word   -1610120712
4353         .word   1728
4354         .word   1443096928
4355         .word   1912
4356         .word   570530944
4357         .word   1592
4358         .word   -2029893008
4359         .word   1544
4360         .word   -654195064
4361         .word   2032
4362         .word   -1945742064
4363         .word   432
4364         .word   -1744396200
4365         .word   1656
4366         .word   -1509446648
4367         .word   320
4368         .word   -1526475024
4369         .word   304
4370         .word   -637158288
4371         .word   1312
4372         .word   1057320440
4373         .word   1824
4374         .word   738317544
4375         .word   104
4376         .word   1342424208
4377         .word   1240
4378         .word   1778581088
4379         .word   784
4380         .word   1409544752
4381         .word   1552
4382         .word   -167483240
4383         .word   1856
4384         .word   -1878604352
4385         .word   752
4386         .word   771870648
4387         .word   1960
4388         .word   -2113528456
4389         .word   1520
4390         .word   -1627198464
4391         .word   992
4392         .word   1762034840
4393         .word   1352
4394         .word   1862707560
4395         .word   1432
4396         .word   -822007664
4397         .word   472
4398         .word   -939170616
4399         .word   1336
4400         .word   268485608
4401         .word   880
4402         .word   -402332904
4403         .word   984
4404         .word   -620634664
4405         .word   72
4406         .word   -855559232
4407         .word   1952
4408         .word   1845676224
4409         .word   8
4410         .word   -335227464
4411         .word   1344
4412         .word   -2096988976
4413         .word   808
4414         .word   -435901584
4415         .word   1008
4416         .word   -1442316496
```

```
4417         .word   64
4418         .word   554034808
4419         .word   1840
4420         .word   -285167808
4421         .word   1736
4422         .word   -1173930792
4423         .word   1648
4424         .word   1241741744
4425         .word   1696
4426         .word   -368773048
4427         .word   1712
4428         .word   688227296
4429         .word   1400
4430         .word   822420880
4431         .word   392
4432         .word   704772376
4433         .word   384
4434         .word   -972739424
4435         .word   1536
4436         .word   889525040
4437         .word   440
4438         .word   1946318304
4439         .word   1328
4440         .word   -66841008
4441         .word   1408
4442         .word   -536574336
4443         .word   168
4444         .word   855981760
4445         .word   592
4446         .word   -251648832
4447         .word   1976
4448         .word   1091004112
4449         .word   112
4450         .word   2131126912
4451         .word   376
4452         .word   386174896
4453         .word   1128
4454         .word   1979870896
4455         .word   616
4456         .word   1124564352
4457         .word   672
4458         .word   -872066456
4459         .word   1784
4460         .word   -469454816
4461         .word   1816
4462         .word   -1643737688
4463         .word   216
4464         .word   1275286592
4465         .word   1472
4466         .word   -1056874248
4467         .word   1016
4468         .word   1174612616
4469         .word   32
4470         .word   -1660750000
4471         .word   744
4472         .word   17064360
4473         .word   920
4474         .word   -100385888
4475         .word   368
4476         .word   -83863032
4477         .word   720
4478         .word   -1291634456
4479         .word   656
4480         .word   -1845043568
4481         .word   408
4482         .word   -385842512
```



```
4483         .word   152
4484         .word   1829155384
4485         .word   1120
4486         .word   -1710834936
4487         .word   976
4488         .word   923076704
4489         .word   1136
4490         .word   1493680288
4491         .word   1096
4492         .word   -352282144
4493         .word   1904
4494         .word   -838514376
4495         .word   424
4496         .word   -1224536504
4497         .word   1896
4498         .word   -520034520
4499         .word   480
4500         .word   2046967176
4501         .word   712
4502         .word   -1677289736
4503         .word   504
4504         .word   1426559896
4505         .word   968
4506         .word   402695792
4507         .word   1528
4508         .word   1929787832
4509         .word   1872
4510         .word   1393016424
4511         .word   728
4512         .word   1594355024
4513         .word   160
4514         .word   -553522312
4515         .word   1072
4516         .word   2013406936
4517         .word   1032
4518         .word   -905609320
4519         .word   496
4520         .word   -1190967776
4521         .word   352
4522         .word   939598240
4523         .word   760
4524         .word   -1039853056
4525         .word   912
4526         .word   369159704
4527         .word   96
4528         .word   -1140387544
4529         .word   1112
4530         .word   671212104
4531         .word   520
4532         .word   -16749400
4533         .word   904
4534         .word   956645384
4535         .word   1776
4536         .word   134243736
4537         .word   1248
4538         .word   -670718176
4539         .word   1152
4540         .word   1677899272
4541         .word   776
4542         .word   2064014368
4543         .word   896
4544         .word   -721316432
4545         .word   928
4546         .word   1208181472
4547         .word   528
4548         .word   -804928840
```

```
4549         .type   Td1,#object
4550         .size   Td1,2048
4551 !
4552 ! CONSTANT POOL
4553 !
4554 Td2:
4555         .word   1336
4556         .word   1342345120
4557         .word   808
4558         .word   1392767496
4559         .word   1312
4560         .word   -1023356744
4561         .word   752
4562         .word   -1778265800
4563         .word   856
4564         .word   -889070248
4565         .word   552
4566         .word   -251593496
4567         .word   704
4568         .word   -1425709104
4569         .word   24
4570         .word   -1828561128
4571         .word   2000
4572         .word   1426129280
4573         .word   872
4574         .word   -167416912
4575         .word   944
4576         .word   -1861990816
4577         .word   608
4578         .word   621258768
4579         .word   1720
4580         .word   -66945240
4581         .word   1624
4582         .word   -687462064
4583         .word   544
4584         .word   -2147405400
4585         .word   1304
4586         .word   -1895453936
4587         .word   720
4588         .word   1225192840
4589         .word   216
4590         .word   1728130512
4591         .word   112
4592         .word   -1744687280
4593         .word   1536
4594         .word   -519901200
4595         .word   936
4596         .word   33954168
4597         .word   1920
4598         .word   302254688
4599         .word   1208
4600         .word   -1559991760
4601         .word   1992
4602         .word   -972857704
4603         .word   760
4604         .word   -419423112
4605         .word   1248
4606         .word   -1795117936
4607         .word   976
4608         .word   -351929496
4609         .word   712
4610         .word   -637228400
4611         .word   1048
4612         .word   755410416
4613         .word   264
4614         .word   -754793568
```



```
4615         .word   840
4616         .word   688017152
4617         .word   1600
4618         .word   1141143112
4619         .word   1096
4620         .word   1778626064
4621         .word   968
4622         .word   2013766768
4623         .word   496
4624         .word   1795476160
4625         .word   904
4626         .word   -587121208
4627         .word   632
4628         .word   -1241123064
4629         .word   1384
4630         .word   386368576
4631         .word   1376
4632         .word   1711687936
4633         .word   464
4634         .word   -1274810768
4635         .word   592
4636         .word   402857720
4637         .word   392
4638         .word   -2113460016
4639         .word   408
4640         .word   1610922632
4641         .word   1016
4642         .word   1157829272
4643         .word   952
4644         .word   -536507616
4645         .word   1392
4646         .word   -2079990952
4647         .word   1280
4648         .word   470283272
4649         .word   344
4650         .word   -1811429312
4651         .word   832
4652         .word   1476624960
4653         .word   2024
4654         .word   419723816
4655         .word   864
4656         .word   -2029738256
4657         .word   1984
4658         .word   -1224567848
4659         .word   1688
4660         .word   587553688
4661         .word   16
4662         .word   -503082408
4663         .word   1144
4664         .word   1460082936
4665         .word   1368
4666         .word   704852648
4667         .word   320
4668         .word   117806936
4669         .word   1552
4670         .word   50429352
4671         .word   984
4672         .word   -1711000024
4673         .word   64
4674         .word   -1526294088
4675         .word   1080
4676         .word   -234782400
4677         .word   1320
4678         .word   -1308549640
4679         .word   848
4680         .word   -1174401000
```

```
4681         .word   1040
4682         .word   1543989424
4683         .word   224
4684         .word   721704568
4685         .word   1440
4686         .word   -1845150776
4687         .word   1936
4688         .word   -267937736
4689         .word   1808
4690         .word   -1593674936
4691         .word   1952
4692         .word   -855429424
4693         .word   1520
4694         .word   -721407960
4695         .word   784
4696         .word   520522144
4697         .word   2032
4698         .word   -1979308752
4699         .word   664
4700         .word   -1660837520
4701         .word   680
4702         .word   -1610279016
4703         .word   1800
4704         .word   838872144
4705         .word   1880
4706         .word   1963272112
4707         .word   1888
4708         .word   956324888
4709         .word   1912
4710         .word   -1442708736
4711         .word   1272
4712         .word   100856712
4713         .word   128
4714         .word   1359342448
4715         .word   1104
4716         .word   -117313272
4717         .word   48
4718         .word   1023719144
4719         .word   40
4720         .word   -1375278608
4721         .word   1512
4722         .word   1174564656
4723         .word   1128
4724         .word   -1257993568
4725         .word   744
4726         .word   84119072
4727         .word   1696
4728         .word   1862279216
4729         .word   168
4730         .word   -16579968
4731         .word   2008
4732         .word   604032192
4733         .word   1864
4734         .word   -1761167896
4735         .word   536
4736         .word   -872134144
4737         .word   1264
4738         .word   1996701384
4739         .word   528
4740         .word   -1123711168
4741         .word   1112
4742         .word   -2013250488
4743         .word   728
4744         .word   939997384
4745         .word   1904
4746         .word   -620507584
```



```
4747         .word   80
4748         .word   1191513056
4749         .word   120
4750         .word   -385621488
4751         .word   240
4752         .word   -922237920
4753         .word   0
4754         .word   0
4755         .word   1072
4756         .word   -2097132544
4757         .word   1896
4758         .word   1208062296
4759         .word   896
4760         .word   -1409224568
4761         .word   912
4762         .word   1308844752
4763         .word   2040
4764         .word   -83367824
4765         .word   448
4766         .word   1442872360
4767         .word   1704
4768         .word   503442800
4769         .word   456
4770         .word   654422376
4771         .word   1736
4772         .word   1677742200
4773         .word   1328
4774         .word   553861856
4775         .word   672
4776         .word   -788210984
4777         .word   368
4778         .word   973152688
4779         .word   824
4780         .word   -1325375408
4781         .word   1848
4782         .word   251959992
4783         .word   1200
4784         .word   -771381392
4785         .word   1160
4786         .word   -1644110632
4787         .word   1576
4788         .word   1325663744
4789         .word   256
4790         .word   -1576857888
4791         .word   600
4792         .word   1761792952
4793         .word   208
4794         .word   369156240
4795         .word   1488
4796         .word   168236184
4797         .word   336
4798         .word   -452590336
4799         .word   1792
4800         .word   1124196624
4801         .word   184
4802         .word   486576344
4803         .word   104
4804         .word   184578120
4805         .word   1592
4806         .word   -1392012200
4807         .word   1344
4808         .word   -1191088720
4809         .word   1352
4810         .word   -939482896
4811         .word   200
4812         .word   -2063417464
```

```
4813         .word   56
4814         .word   1275427752
4815         .word   1768
4816         .word   -1157139256
4817         .word   768
4818         .word   -49996808
4819         .word   304
4820         .word   -1626884088
4821         .word   1960
4822         .word   -1140661360
4823         .word   472
4824         .word   -989715664
4825         .word   1008
4826         .word   872603608
4827         .word   328
4828         .word   1979996696
4829         .word   1584
4830         .word   -603563752
4831         .word   2016
4832         .word   1745205096
4833         .word   1928
4834         .word   1661323040
4835         .word   1760
4836         .word   -905528952
4837         .word   1064
4838         .word   268571416
4839         .word   272
4840         .word   1073781944
4841         .word   136
4842         .word   537142832
4843         .word   288
4844         .word   2097424976
4845         .word   488
4846         .word   -133786152
4847         .word   400
4848         .word   285571016
4849         .word   1288
4850         .word   1829124424
4851         .word   376
4852         .word   1258351856
4853         .word   384
4854         .word   -217651824
4855         .word   656
4856         .word   -335516624
4857         .word   1816
4858         .word   -805061112
4859         .word   176
4860         .word   1812028824
4861         .word   1480
4862         .word   -1727706240
4863         .word   576
4864         .word   -100627296
4865         .word   800
4866         .word   570572616
4867         .word   1120
4868         .word   -1006286880
4869         .word   504
4870         .word   436537216
4871         .word   352
4872         .word   -670911512
4873         .word   1152
4874         .word   -285142632
4875         .word   624
4876         .word   -956024248
4877         .word   1672
4878         .word   -1056519744
```



```
4879         .word   1296
4880         .word   -33266096
4881         .word   88
4882         .word   906282656
4883         .word   1032
4884         .word   -821741656
4885         .word   1776
4886         .word   671427536
4887         .word   1136
4888         .word   637982136
4889         .word   1528
4890         .word   -1543373464
4891         .word   1256
4892         .word   -469671472
4893         .word   1168
4894         .word   218268608
4895         .word   1632
4896         .word   -1694280968
4897         .word   560
4898         .word   1644340208
4899         .word   152
4900         .word   -1039682456
4901         .word   1472
4902         .word   -402356544
4903         .word   1976
4904         .word   1577152968
4905         .word   1400
4906         .word   -184281576
4907         .word   1024
4908         .word   -1106969880
4909         .word   1176
4910         .word   2080591488
4911         .word   360
4912         .word   -1459388760
4913         .word   144
4914         .word   -1291421400
4915         .word   1224
4916         .word   990266720
4917         .word   1000
4918         .word   -1493139264
4919         .word   792
4920         .word   1845970144
4921         .word   1496
4922         .word   2064046552
4923         .word   960
4924         .word   151415088
4925         .word   192
4926         .word   -201100600
4927         .word   1464
4928         .word   17261776
4929         .word   1232
4930         .word   -1476126088
4931         .word   880
4932         .word   1694971048
4933         .word   1840
4934         .word   2114279416
4935         .word   1656
4936         .word   134286816
4937         .word   1856
4938         .word   -435717976
4939         .word   1240
4940         .word   -653928648
4941         .word   432
4942         .word   -838708360
4943         .word   72
4944         .word   -737717000
```

```
4945         .word   992
4946         .word   -704557696
4947         .word   1424
4948         .word   -1358852832
4949         .word   280
4950         .word   822170104
4951         .word   1184
4952         .word   805713192
4953         .word   816
4954         .word   -1073631984
4955         .word   1504
4956         .word   922985072
4957         .word   1616
4958         .word   -1509432304
4959         .word   1664
4960         .word   -1341717376
4961         .word   1728
4962         .word   352427320
4963         .word   1216
4964         .word   1242007584
4965         .word   1744
4966         .word   -150859936
4967         .word   640
4968         .word   235142760
4969         .word   1968
4970         .word   788577416
4971         .word   1712
4972         .word   -1929137560
4973         .word   1408
4974         .word   1291984760
4975         .word   616
4976         .word   1409705296
4977         .word   32
4978         .word   -553179984
4979         .word   1448
4980         .word   -486214008
4981         .word   1088
4982         .word   453141328
4983         .word   248
4984         .word   -1207563936
4985         .word   648
4986         .word   2130850600
4987         .word   1872
4988         .word   67431152
4989         .word   424
4990         .word   1560284256
4991         .word   928
4992         .word   1929892920
4993         .word   520
4994         .word   772266072
4995         .word   232
4996         .word   1510316856
4997         .word   1680
4998         .word   1376032472
4999         .word   688
5000         .word   856115328
5001         .word   568
5002         .word   318992048
5003         .word   776
5004         .word   -1945839944
5005         .word   96
5006         .word   2046934280
5007         .word   160
5008         .word   -1912418368
5009         .word   480
5010         .word   -1996007272
```



```
5011         .word   312
5012         .word   -301566648
5013         .word   1608
5014         .word   889568008
5015         .word   1832
5016         .word   -318306080
5017         .word   1416
5018         .word   1006883384
5019         .word   1784
5020         .word   1493493392
5021         .word   920
5022         .word   1057140624
5023         .word   1648
5024         .word   2030092448
5025         .word   440
5026         .word   -1090281928
5027         .word   1640
5028         .word   -368926792
5029         .word   1360
5030         .word   1526923240
5031         .word   888
5032         .word   336001512
5033         .word   1752
5034         .word   -2046574048
5035         .word   1944
5036         .word   -2130291336
5037         .word   1568
5038         .word   1040567104
5039         .word   416
5040         .word   738312480
5041         .word   512
5042         .word   1594234136
5043         .word   1560
5044         .word   1912647912
5045         .word   296
5046         .word   201713424
5047         .word   584
5048         .word   -1962851872
5049         .word   1192
5050         .word   1091041384
5051         .word   8
5052         .word   1895943488
5053         .word   1432
5054         .word   -570408864
5055         .word   1824
5056         .word   -1677277792
5057         .word   1544
5058         .word   -1878842704
5059         .word   1056
5060         .word   1627643480
5061         .word   1456
5062         .word   1879484816
5063         .word   736
5064         .word   1946305376
5065         .word   696
5066         .word   1107723712
5067         .type   Td2,#object
5068         .size   Td2,2048
5069 !
5070 ! CONSTANT POOL
5071 !
5072 Td3:
5073         .word   1952
5074         .word   -1493007736
5075         .word   520
5076         .word   1694669808
```

```
5077         .word   184
5078         .word   -1543104304
5079         .word   312
5080         .word   1577365968
5081         .word   1368
5082         .word   1795578328
5083         .word   1256
5084         .word   1158121720
5085         .word   2000
5086         .word   1476746592
5087         .word   1816
5088         .word   50633304
5089         .word   384
5090         .word   -100488960
5091         .word   944
5092         .word   1829221736
5093         .word   1632
5094         .word   1980009536
5095         .word   16
5096         .word   1275146152
5097         .word   1832
5098         .word   -687349128
5099         .word   336
5100         .word   -888750552
5101         .word   424
5102         .word   1141113136
5103         .word   784
5104         .word   -1559986776
5105         .word   1416
5106         .word   1510100720
5107         .word   1488
5108         .word   453196072
5109         .word   1872
5110         .word   235192872
5111         .word   2032
5112         .word   -1073280280
5113         .word   376
5114         .word   1962939928
5115         .word   608
5116         .word   -268397560
5117         .word   560
5118         .word   -1761272728
5119         .word   1688
5120         .word   -117034152
5121         .word   1144
5122         .word   1594308632
5123         .word   1168
5124         .word   -1677416280
5125         .word   872
5126         .word   2047303160
5127         .word   656
5128         .word   1493619880
5129         .word   1520
5130         .word   -2097058144
5131         .word   928
5132         .word   554080960
5133         .word   1792
5134         .word   1761692232
5135         .word   1608
5136         .word   -939383696
5137         .word   1552
5138         .word   -1996270680
5139         .word   1136
5140         .word   2030290848
5141         .word   704
5142         .word   1040407752
```



```
5143         .word   1480
5144         .word   1896278328
5145         .word   1800
5146         .word   1325774320
5147         .word   1088
5148         .word   -1392459904
5149         .word   256
5150         .word   -1409075640
5151         .word   1648
5152         .word   973448168
5153         .word   1784
5154         .word   1241563928
5155         .word   208
5156         .word   822351656
5157         .word   648
5158         .word   855835832
5159         .word   664
5160         .word   2130848528
5161         .word   800
5162         .word   1996948872
5163         .word   856
5164         .word   -1375459880
5165         .word   1032
5166         .word   -1610553360
5167         .word   64
5168         .word   721725384
5169         .word   576
5170         .word   1745011584
5171         .word   552
5172         .word   -50279304
5173         .word   1776
5174         .word   1812216992
5175         .word   984
5176         .word   -133842288
5177         .word   920
5178         .word   -754901672
5179         .word   600
5180         .word   34018192
5181         .word   248
5182         .word   -1895645416
5183         .word   680
5184         .word   -1425976528
5185         .word   1880
5186         .word   671104400
5187         .word   1448
5188         .word   -1040180872
5189         .word   1576
5190         .word   2063914032
5191         .word   440
5192         .word   134557336
5193         .word   320
5194         .word   -2029547136
5195         .word   1528
5196         .word   -1526361832
5197         .word   24
5198         .word   1778765840
5199         .word   176
5200         .word   -2113738904
5201         .word   1656
5202         .word   469851216
5203         .word   968
5204         .word   -1274768072
5205         .word   56
5206         .word   -234387560
5207         .word   840
5208         .word   -502986128
```

```
5209          .word   1744
5210          .word   -200905944
5211          .word   40
5212          .word   -1106859984
5213          .word   416
5214          .word   1644232328
5215          .word   1328
5216          .word   -33270240
5217          .word   368
5218          .word   1392830880
5219          .word   1944
5220          .word   1426392336
5221          .word   1104
5222          .word   -519991256
5223          .word   1968
5224          .word   -352080608
5225          .word   1048
5226          .word   -335427496
5227          .word   768
5228          .word   -284864000
5229          .word   904
5230          .word   -1627376912
5231          .word   880
5232          .word   268602856
5233          .word   264
5234          .word   -1979201040
5235          .word   1768
5236          .word   100789424
5237          .word   496
5238          .word   84244200
5239          .word   1840
5240          .word   -1123929496
5241          .word   672
5242          .word   -1929007992
5243          .word   1568
5244          .word   1560292232
5245          .word   48
5246          .word   -737970144
5247          .word   640
5248          .word   352844544
5249          .word   1216
5250          .word   -83812152
5251          .word   1512
5252          .word   -385565008
5253          .word   512
5254          .word   1124492360
5255          .word   1736
5256          .word   -1643922632
5257          .word   1856
5258          .word   1107684736
5259          .word   1096
5260          .word   -1962655688
5261          .word   200
5262          .word   1526843192
5263          .word   1600
5264          .word   -301540408
5265          .word   992
5266          .word   167918856
5267          .word   528
5268          .word   252136416
5269          .word   1056
5270          .word   503730112
5271          .word   0
5272          .word   0
5273          .word   1024
5274          .word   -2046551992
```



```
5275          .word   344
5276          .word   -318619248
5277          .word   136
5278          .word   1879400688
5279          .word   720
5280          .word   1912763232
5281          .word   112
5282          .word   -16261144
5283          .word   1064
5284          .word   939700344
5285          .word   1392
5286          .word   -721358360
5287          .word   360
5288          .word   956381616
5289          .word   120
5290          .word   -654106544
5291          .word   736
5292          .word   -1509881024
5293          .word   728
5294          .word   1409715416
5295          .word   432
5296          .word   771871008
5297          .word   80
5298          .word   1728415840
5299          .word   696
5300          .word   -419398504
5301          .word   1904
5302          .word   -1777953376
5303          .word   1240
5304          .word   -1861947176
5305          .word   1536
5306          .word   -989692928
5307          .word   1760
5308          .word   537203464
5309          .word   952
5310          .word   1258506960
5311          .word   144
5312          .word   436252896
5313          .word   1176
5314          .word   -1174382832
5315          .word   1280
5316          .word   705113600
5317          .word   272
5318          .word   -536733216
5319          .word   216
5320          .word   385935504
5321          .word   72
5322          .word   218126448
5323          .word   1112
5324          .word   -955945072
5325          .word   1456
5326          .word   -1476015768
5327          .word   240
5328          .word   -1459208032
5329          .word   1928
5330          .word   419703480
5331          .word   936
5332          .word   117597560
5333          .word   1224
5334          .word   -586817680
5335          .word   1016
5336          .word   1611132184
5337          .word   8
5338          .word   637861816
5339          .word   912
5340          .word   -184163616
```

```
5341         .word   816
5342         .word   990259744
5343         .word   2008
5344         .word   2114036440
5345         .word   536
5346         .word   688108632
5347         .word   280
5348         .word   -972626344
5349         .word   1896
5350         .word   -66894416
5351         .word   1824
5352         .word   -251454016
5353         .word   392
5354         .word   -603564360
5355         .word   792
5356         .word   -2063564272
5357         .word   1208
5358         .word   570556568
5359         .word   1584
5360         .word   285279264
5361         .word   592
5362         .word   604236840
5363         .word   1496
5364         .word   1023919760
5365         .word   1992
5366         .word   838897008
5367         .word   328
5368         .word   -1593610696
5369         .word   1264
5370         .word   788682984
5371         .word   1424
5372         .word   805805792
5373         .word   1072
5374         .word   1376215144
5375         .word   1544
5376         .word   -486112328
5377         .word   1432
5378         .word   369320280
5379         .word   896
5380         .word   -1190867640
5381         .word   1184
5382         .word   1208471688
5383         .word   1864
5384         .word   1677791800
5385         .word   2016
5386         .word   -1945754304
5387         .word   1920
5388         .word   1057019136
5389         .word   1000
5390         .word   738640560
5391         .word   408
5392         .word   -1878558448
5393         .word   584
5394         .word   1309031480
5395         .word   448
5396         .word   -788132152
5397         .word   1616
5398         .word   -1576536992
5399         .word   1696
5400         .word   184661184
5401         .word   1960
5402         .word   -2130281168
5403         .word   976
5404         .word   -570342104
5405         .word   1464
5406         .word   -1912523056
```



```
5407         .word   1384
5408         .word   -1090182664
5409         .word   464
5410         .word   -1660477088
5411         .word   960
5412         .word   -1845466496
5413         .word   760
5414         .word   -872096944
5415         .word   1008
5416         .word   1174606496
5417         .word   1128
5418         .word   319166384
5419         .word   1728
5420         .word   -1207483264
5421         .word   456
5422         .word   -150802064
5423         .word   1560
5424         .word   -1358451696
5425         .word   744
5426         .word   -2147093256
5427         .word   1664
5428         .word   -1828461752
5429         .word   1704
5430         .word   755321720
5431         .word   296
5432         .word   302358136
5433         .word   1376
5434         .word   -1727930816
5435         .word   192
5436         .word   2097494144
5437         .word   1248
5438         .word   1661171520
5439         .word   472
5440         .word   -1157374248
5441         .word   304
5442         .word   2013285992
5443         .word   712
5444         .word   403153776
5445         .word   1232
5446         .word   -1224732832
5447         .word   632
5448         .word   -1710930920
5449         .word   1192
5450         .word   1845702448
5451         .word   2040
5452         .word   -435948208
5453         .word   1504
5454         .word   -822066936
5455         .word   168
5456         .word   -402180232
5457         .word   1848
5458         .word   -1694052912
5459         .word   888
5460         .word   906392144
5461         .word   1272
5462         .word   151430992
5463         .word   1408
5464         .word   2080813384
5465         .word   1312
5466         .word   -1308264056
5467         .word   504
5468         .word   587303248
5469         .word   1320
5470         .word   -1811839440
5471         .word   1296
5472         .word   1711669672
```

```
5473         .word   624
5474         .word   -1140737120
5475         .word   1040
5476         .word   -905627680
5477         .word   1152
5478         .word   -804944128
5479         .word   1336
5480         .word   -671045224
5481         .word   32
5482         .word   -1744676984
5483         .word   1888
5484         .word   -637027832
5485         .word   1640
5486         .word   1342206968
5487         .word   1160
5488         .word   -167675720
5489         .word   616
5490         .word   -704353360
5491         .word   1912
5492         .word   -1342019048
5493         .word   1360
5494         .word   1292019296
5495         .word   1200
5496         .word   67567392
5497         .word   1672
5498         .word   -1257825040
5499         .word   848
5500         .word   -2013210016
5501         .word   352
5502         .word   520472072
5503         .word   808
5504         .word   1359215152
5505         .word   752
5506         .word   -369089304
5507         .word   1120
5508         .word   889382920
5509         .word   1080
5510         .word   1946394576
5511         .word   88
5512         .word   1090615256
5513         .word   824
5514         .word   486725016
5515         .word   1752
5516         .word   -771582832
5517         .word   128
5518         .word   1442946888
5519         .word   1712
5520         .word   1191222120
5521         .word   1720
5522         .word   1627677904
5523         .word   1288
5524         .word   201576888
5525         .word   1984
5526         .word   335835848
5527         .word   152
5528         .word   1006915416
5529         .word   1352
5530         .word   654800496
5531         .word   776
5532         .word   -922636872
5533         .word   224
5534         .word   -452497656
5535         .word   568
5536         .word   -1325276208
5537         .word   1680
5538         .word   -553464608
```



```
5539         .word   1936
5540         .word   1929509544
5541         .word   160
5542         .word   -838612800
5543         .word   1592
5544         .word   923138968
5545         .word   1976
5546         .word   -855158120
5547         .word   2024
5548         .word   -1442653448
5549         .word   488
5550         .word   1862313720
5551         .word   544
5552         .word   -620481600
5553         .word   1400
5554         .word   -217838000
5555         .word   832
5556         .word   -1006504504
5557         .word   288
5558         .word   872505792
5559         .word   1304
5560         .word   1073937936
5561         .word   232
5562         .word   -1023176528
5563         .word   1808
5564         .word   620783072
5565         .word   480
5566         .word   1225021760
5567         .word   104
5568         .word   -1795026952
5569         .word   1344
5570         .word   17009096
5571         .word   96
5572         .word   -1291390912
5573         .word   1440
5574         .word   -469440832
5575         .word   688
5576         .word   -1056668896
5577         .word   1624
5578         .word   -2080175144
5579         .word   400
5580         .word   -1241282904
5581         .word   864
5582         .word   1543742016
5583         .word   1472
5584         .word   1459754624
5585         .type   Td3,#object
5586         .size   Td3,2048
5587         .align  4
5588 !
5589 ! CONSTANT POOL
5590 !
5591 Td4:
5592         .word   1381126738
5593         .word   151587081
5594         .word   1785358954
5595         .word   -707406379
5596         .word   808464432
5597         .word   909522486
5598         .word   -1515870811
5599         .word   943208504
5600         .word   -1077952577
5601         .word   1077952576
5602         .word   -1549556829
5603         .word   -1633771874
5604         .word   -2122219135
```

```
5605          .word   -202116109
5606          .word   -673720361
5607          .word   -67372037
5608          .word   2088533116
5609          .word   -471604253
5610          .word   960051513
5611          .word   -2105376126
5612          .word   -1684300901
5613          .word   791621423
5614          .word   -1
5615          .word   -2021161081
5616          .word   875836468
5617          .word   -1903260018
5618          .word   1128481603
5619          .word   1145324612
5620          .word   -993737532
5621          .word   -555819298
5622          .word   -370546199
5623          .word   -875836469
5624          .word   1414812756
5625          .word   2071690107
5626          .word   -1802201964
5627          .word   842150450
5628          .word   -1499027802
5629          .word   -1027423550
5630          .word   589505315
5631          .word   1027423549
5632          .word   -286331154
5633          .word   1280068684
5634          .word   -1785358955
5635          .word   185273099
5636          .word   1111638594
5637          .word   -84215046
5638          .word   -1010580541
5639          .word   1313754702
5640          .word   134744072
5641          .word   774778414
5642          .word   -1583242847
5643          .word   1717986918
5644          .word   673720360
5645          .word   -640034343
5646          .word   606348324
5647          .word   -1296911694
5648          .word   1987475062
5649          .word   1532713819
5650          .word   -1566399838
5651          .word   1229539657
5652          .word   1835887981
5653          .word   -1953789045
5654          .word   -774778415
5655          .word   623191333
5656          .word   1920103026
5657          .word   -117901064
5658          .word   -151587082
5659          .word   1684300900
5660          .word   -2038004090
5661          .word   1751672936
5662          .word   -1734829928
5663          .word   370546198
5664          .word   -724249388
5665          .word   -1532713820
5666          .word   1549556828
5667          .word   -858993460
5668          .word   1566399837
5669          .word   1701143909
5670          .word   -1229539658
```



```
5671          .word   -1835887982
5672          .word   1819044972
5673          .word   1886417008
5674          .word   1212696648
5675          .word   1347440720
5676          .word   -33686019
5677          .word   -303174163
5678          .word   -1179010631
5679          .word   -623191334
5680          .word   1583242846
5681          .word   353703189
5682          .word   1179010630
5683          .word   1465341783
5684          .word   -1482184793
5685          .word   -1920103027
5686          .word   -1650614883
5687          .word   -2071690108
5688          .word   -1869574000
5689          .word   -656877352
5690          .word   -1414812757
5691          .word   0
5692          .word   -1936946036
5693          .word   -1128481604
5694          .word   -741092397
5695          .word   168430090
5696          .word   -134744073
5697          .word   -454761244
5698          .word   1482184792
5699          .word   84215045
5700          .word   -1195853640
5701          .word   -1280068685
5702          .word   1162167621
5703          .word   101058054
5704          .word   -791621424
5705          .word   741092396
5706          .word   505290270
5707          .word   -1886417009
5708          .word   -892679478
5709          .word   1061109567
5710          .word   252645135
5711          .word   33686018
5712          .word   -1044266559
5713          .word   -1347440721
5714          .word   -1111638595
5715          .word   50529027
5716          .word   16843009
5717          .word   320017171
5718          .word   -1970632054
5719          .word   1802201963
5720          .word   976894522
5721          .word   -1852730991
5722          .word   286331153
5723          .word   1094795585
5724          .word   1330597711
5725          .word   1734829927
5726          .word   -589505316
5727          .word   -353703190
5728          .word   -1751672937
5729          .word   -218959118
5730          .word   -808464433
5731          .word   -825307442
5732          .word   -252645136
5733          .word   -1263225676
5734          .word   -421075226
5735          .word   1936946035
5736          .word   -1768515946
```

```
5737         .word   -1397969748
5738         .word   1953789044
5739         .word   572662306
5740         .word   -404232217
5741         .word   -1381126739
5742         .word   892679477
5743         .word   -2054847099
5744         .word   -488447262
5745         .word   -101058055
5746         .word   926365495
5747         .word   -387389208
5748         .word   471604252
5749         .word   1970632053
5750         .word   -538976289
5751         .word   1852730990
5752         .word   1195853639
5753         .word   -235802127
5754         .word   437918234
5755         .word   1903260017
5756         .word   488447261
5757         .word   690563369
5758         .word   -976894523
5759         .word   -1987475063
5760         .word   1869573999
5761         .word   -1212696649
5762         .word   1650614882
5763         .word   235802126
5764         .word   -1431655766
5765         .word   404232216
5766         .word   -1094795586
5767         .word   454761243
5768         .word   -50529028
5769         .word   1448498774
5770         .word   1044266558
5771         .word   1263225675
5772         .word   -960051514
5773         .word   -757935406
5774         .word   2038004089
5775         .word   538976288
5776         .word   -1701143910
5777         .word   -606348325
5778         .word   -1061109568
5779         .word   -16843010
5780         .word   2021161080
5781         .word   -842150451
5782         .word   1515870810
5783         .word   -185273100
5784         .word   522133279
5785         .word   -572662307
5786         .word   -1465341784
5787         .word   858993459
5788         .word   -2004318072
5789         .word   117901063
5790         .word   -943208505
5791         .word   825307441
5792         .word   -1313754703
5793         .word   303174162
5794         .word   269488144
5795         .word   1499027801
5796         .word   656877351
5797         .word   -2139062144
5798         .word   -320017172
5799         .word   1600085855
5800         .word   1616928864
5801         .word   1364283729
5802         .word   2139062143
```



```
5803         .word   -1448498775
5804         .word   421075225
5805         .word   -1246382667
5806         .word   1246382666
5807         .word   218959117
5808         .word   757935405
5809         .word   -437918235
5810         .word   2054847098
5811         .word   -1616928865
5812         .word   -1819044973
5813         .word   -909522487
5814         .word   -1667457892
5815         .word   -269488145
5816         .word   -1600085856
5817         .word   -522133280
5818         .word   993737531
5819         .word   1296911693
5820         .word   -1364283730
5821         .word   707406378
5822         .word   -168430091
5823         .word   -1330597712
5824         .word   -926365496
5825         .word   -336860181
5826         .word   -1145324613
5827         .word   1010580540
5828         .word   -2088533117
5829         .word   1397969747
5830         .word   -1717986919
5831         .word   1633771873
5832         .word   387389207
5833         .word   724249387
5834         .word   67372036
5835         .word   2122219134
5836         .word   -1162167622
5837         .word   2004318071
5838         .word   -690563370
5839         .word   640034342
5840         .word   -505290271
5841         .word   1768515945
5842         .word   336860180
5843         .word   1667457891
5844         .word   1431655765
5845         .word   555819297
5846         .word   202116108
5847         .word   2105376125
5848         .type   Td4,#object
5849         .size   Td4,1024

5851 !
5852 ! CONSTANT POOL
5853 !
5854 rcon:
5855         .word   16777216
5856         .word   33554432
5857         .word   67108864
5858         .word   134217728
5859         .word   268435456
5860         .word   536870912
5861         .word   1073741824
5862         .word   -2147483648
5863         .word   452984832
5864         .word   905969664
5865         .type   rcon,#object
5866         .size   rcon,40

5870 /* EXPORT DELETE END */
```

```
5868 ! Begin Disassembling Stabs
5869         .xstabs ".stab.index","Xa ; O ; P ; V=3.1 ; R=WorkShop Compilers 5.0 99/
5870 ! End Disassembling Stabs

5872 ! Begin Disassembling Ident
5873         .ident  "cg: WorkShop Compilers 5.0 99/04/15 Compiler Common 5.0 Patch 1
5874         .ident  "acomp: WorkShop Compilers 5.0 99/02/25 C 5.0 patch 107289-01"
5875 ! End Disassembling Ident

5877 #endif  /* lint || __lint */
```

```
**********************************************************
    1015 Thu Jul 11 01:29:01 2013
new/usr/src/common/crypto/arcfour/Makefile
first pass
**********************************************************
```

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

.KEEP_STATE:

FRC:

# EXPORT DELETE START
EXPORT_SRC:
        $(RM) Makefile+ arcfour_crypt.c+ amd64/arcfour-x86_64.pl+ \
                sun4u/arcfour_crypt_asm.s+ sun4v/arcfour_crypt.c+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < arcfour_crypt.c > arcfour_crypt.c+
        $(MV) arcfour_crypt.c+ arcfour_crypt.c
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < amd64/arcfour-x86_64.pl > amd64/arcfour-x86_64.pl+
        $(MV) amd64/arcfour-x86_64.pl+ amd64/arcfour-x86_64.pl
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < sun4u/arcfour_crypt_asm.s > sun4u/arcfour_crypt_asm.s+
        $(MV) sun4u/arcfour_crypt_asm.s+ sun4u/arcfour_crypt_asm.s
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < sun4v/arcfour_crypt.c > sun4v/arcfour_crypt.c+
        $(MV) sun4v/arcfour_crypt.c+ sun4v/arcfour_crypt.c
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(RM) Makefile
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile arcfour_crypt.c amd64/arcfour-x86_64.pl \
                sun4u/arcfour_crypt_asm.s sun4v/arcfour_crypt.c

# EXPORT DELETE END
```

```
**********************************************************
    9080 Thu Jul 11 01:29:02 2013
new/usr/src/common/crypto/arcfour/amd64/arcfour-x86_64.pl
first pass
**********************************************************
#!/usr/bin/env perl
#
# ====================================================================
# Written by Andy Polyakov <appro@fy.chalmers.se> for the OpenSSL
# project. The module is, however, dual licensed under OpenSSL and
# CRYPTOGAMS licenses depending on where you obtain it. For further
# details see http://www.openssl.org/~appro/cryptogams/.
# ====================================================================
#
# 2.22x RC4 tune-up:-) It should be noted though that my hand [as in
# "hand-coded assembler"] doesn't stand for the whole improvement
# coefficient. It turned out that eliminating RC4_CHAR from config
# line results in ~40% improvement (yes, even for C implementation).
# Presumably it has everything to do with AMD cache architecture and
# RAW or whatever penalties. Once again! The module *requires* config
# line *without* RC4_CHAR! As for coding "secret," I bet on partial
# register arithmetics. For example instead of 'inc %r8; and $255,%r8'
# I simply 'inc %r8b'. Even though optimization manual discourages
# to operate on partial registers, it turned out to be the best bet.
# At least for AMD... How IA32E would perform remains to be seen...

# As was shown by Marc Bevand reordering of couple of load operations
# results in even higher performance gain of 3.3x:-) At least on
# Opteron... For reference, 1x in this case is RC4_CHAR C-code
# compiled with gcc 3.3.2, which performs at ~54MBps per 1GHz clock.
# Latter means that if you want to *estimate* what to expect from
# *your* Opteron, then multiply 54 by 3.3 and clock frequency in GHz.

# Intel P4 EM64T core was found to run the AMD64 code really slow...
# The only way to achieve comparable performance on P4 was to keep
# RC4_CHAR. Kind of ironic, huh? As it's apparently impossible to
# compose blended code, which would perform even within 30% marginal
# on either AMD and Intel platforms, I implement both cases. See
# rc4_skey.c for further details...

# P4 EM64T core appears to be "allergic" to 64-bit inc/dec. Replacing
# those with add/sub results in 50% performance improvement of folded
# loop...

# As was shown by Zou Nanhai loop unrolling can improve Intel EM64T
# performance by >30% [unlike P4 32-bit case that is]. But this is
# provided that loads are reordered even more aggressively! Both code
# pathes, AMD64 and EM64T, reorder loads in essentially same manner
# as my IA-64 implementation. On Opteron this resulted in modest 5%
# improvement [I had to test it], while final Intel P4 performance
# achieves respectful 432MBps on 2.8GHz processor now. For reference.
# If executed on Xeon, current RC4_CHAR code-path is 2.7x faster than
# RC4_INT code-path. While if executed on Opteron, it's only 25%
# slower than the RC4_INT one [meaning that if CPU ´µ-arch detection
# is not implemented, then this final RC4_CHAR code-path should be
# preferred, as it provides better *all-round* performance].

# Intel Core2 was observed to perform poorly on both code paths:-( It
# apparently suffers from some kind of partial register stall, which
# occurs in 64-bit mode only [as virtually identical 32-bit loop was
# observed to outperform 64-bit one by almost 50%]. Adding two movzb to
# cloop1 boosts its performance by 80%! This loop appears to be optimal
# fit for Core2 and therefore the code was modified to skip cloop8 on
# this CPU.

#
```



```
# OpenSolaris OS modifications
#
# Sun elects to use this software under the BSD license.
#
# This source originates from OpenSSL file rc4-x86_64.pl at
# ftp://ftp.openssl.org/snapshot/openssl-0.9.8-stable-SNAP-20080131.tar.gz
# (presumably for future OpenSSL release 0.9.8h), with these changes:
#
# 1. Added some comments, "use strict", and declared all variables.
#
# 2. Added OpenSolaris ENTRY_NP/SET_SIZE macros from
# /usr/include/sys/asm_linkage.h.
#
# 3. Changed function name from RC4() to arcfour_crypt_asm() and RC4_set_key()
# to arcfour_key_init(), and changed the parameter order for both to that
# used by OpenSolaris.
#
# 4. The current method of using cpuid feature bits 20 (NX) or 28 (HTT) from
# function OPENSSL_ia32_cpuid() to distinguish Intel/AMD does not work for
# some newer AMD64 processors, as these bits are set on both Intel EM64T
# processors and newer AMD64 processors.  I replaced this with C code
# (function arcfour_crypt_on_intel()) to call cpuid_getvendor()
# when executing in the kernel and getisax() when executing in userland.
#
# 5. Set a new field in the key structure, key->flag to 0 for AMD AMD64
# and 1 for Intel EM64T.  This is to select the most-efficient arcfour_crypt()
# function to use.
#
# 6. Removed x86_64-xlate.pl script (not needed for as(1) or gas(1) assemblers).
#
# 7. Removed unused RC4_CHAR, Lcloop1, and Lcloop8 code.
#
# 8. Added C function definitions for use by lint(1B).
#

use strict;
my ($code, $dat, $inp, $out, $len, $idx, $ido, $i, @XX, @TX, $YY, $TY);
my $output = shift;
open STDOUT,">$output";

#
# Parameters
#

# OpenSSL:
# void RC4(RC4_KEY *key, unsigned long len, const unsigned char *indata,
#       unsigned char *outdata);
#$dat="%rdi";       # arg1
#$len="%rsi";       # arg2
#$inp="%rdx";       # arg3
#$out="%rcx";       # arg4

# OpenSolaris:
# void arcfour_crypt_asm(ARCFour_key *key, uchar_t *in, uchar_t *out,
#       size_t len);
$dat="%rdi";        # arg1
$inp="%rsi";        # arg2
$out="%rdx";        # arg3
$len="%rcx";        # arg4

#
# Register variables
#
# $XX[0] is key->i (aka key->x), $XX[1] is a temporary.
# $TX[0] and $TX[1] are temporaries.
# $YY is key->j (aka key->y).
```

```
# $TY is a temporary.
#
@XX=("%r8","%r10");
@TX=("%r9","%r11");
$YY="%r12";
$TY="%r13";

$code=<<___;
#if defined(lint) || defined(__lint)

#include "arcfour.h"

/* ARGSUSED */
void
arcfour_crypt_asm(ARCFour_key *key, uchar_t *in, uchar_t *out, size_t len)
{}

/* ARGSUSED */
void
arcfour_key_init(ARCFour_key *key, uchar_t *keyval, int keyvallen)
{}

#else
#include <sys/asm_linkage.h>

ENTRY_NP(arcfour_crypt_asm)
        /* EXPORT DELETE START */

        or      $len,$len # If (len == 0) return
        jne     .Lentry
        ret
.Lentry:
        push    %r12
        push    %r13

        / Set $dat to beginning of array, key->arr[0]
        add     \$8,$dat
        / Get key->j
        movl    -8($dat),$XX[0]#d
        / Get key->i
        movl    -4($dat),$YY#d

        /
        / Use a 4-byte key schedule element array
        /
        inc     $XX[0]#b
        movl    ($dat,$XX[0],4),$TX[0]#d
        test    \$-8,$len
        jz      .Lloop1
        jmp     .Lloop8

.align  16
.Lloop8:
___
for ($i=0;$i<8;$i++) {
$code.=<<___;
        add     $TX[0]#b,$YY#b
        mov     $XX[0],$XX[1]
        movl    ($dat,$YY,4),$TY#d
        ror     \$8,%rax                        # ror is redundant when $i=0
        inc     $XX[1]#b
        movl    ($dat,$XX[1],4),$TX[1]#d
        cmp     $XX[1],$YY
        movl    $TX[0]#d,($dat,$YY,4)
        cmove   $TX[0],$TX[1]
        movl    $TY#d,($dat,$XX[0],4)
```



```
        add     $TX[0]#b,$TY#b
        movb    ($dat,$TY,4),%al
___
push(@TX,shift(@TX)); push(@XX,shift(@XX));     # "rotate" registers
}
$code.=<<___;
        ror     \$8,%rax
        sub     \$8,$len

        xor     ($inp),%rax
        add     \$8,$inp
        mov     %rax,($out)
        add     \$8,$out

        test    \$-8,$len
        jnz     .Lloop8
        cmp     \$0,$len
        jne     .Lloop1

.Lexit:
        /
        / Cleanup and exit code
        /
        / --i to undo ++i done at entry
        sub     \$1,$XX[0]#b
        / set key->i
        movl    $XX[0]#d,-8($dat)
        / set key->j
        movl    $YY#d,-4($dat)

        pop     %r13
        pop     %r12
        ret

.align  16
.Lloop1:
        add     $TX[0]#b,$YY#b
        movl    ($dat,$YY,4),$TY#d
        movl    $TX[0]#d,($dat,$YY,4)
        movl    $TY#d,($dat,$XX[0],4)
        add     $TY#b,$TX[0]#b
        inc     $XX[0]#b
        movl    ($dat,$TX[0],4),$TY#d
        movl    ($dat,$XX[0],4),$TX[0]#d
        xorb    ($inp),$TY#b
        inc     $inp
        movb    $TY#b,($out)
        inc     $out
        dec     $len
        jnz     .Lloop1
        jmp     .Lexit

        /* EXPORT DELETE END */
        ret
SET_SIZE(arcfour_crypt_asm)
___


#
# Parameters
#

# OpenSSL:
# void RC4_set_key(RC4_KEY *key, int len, const unsigned char *data);
#$dat="%rdi";       # arg1
#$len="%rsi";       # arg2
```

```
#$inp="%rdx";       # arg3

# OpenSolaris:
# void arcfour_key_init(ARCFour_key *key, uchar_t *keyval, int keyvallen);
$dat="%rdi";        # arg1
$inp="%rsi";        # arg2
$len="%rdx";        # arg3

# Temporaries
$idx="%r8";
$ido="%r9";

$code.=<<___;
        / int arcfour_crypt_on_intel(void);
.extern arcfour_crypt_on_intel

ENTRY_NP(arcfour_key_init)
        /* EXPORT DELETE START */

        / Find out if we're running on Intel or something else (e.g., AMD64).
        / This sets %eax to 1 for Intel, otherwise 0.
        push    %rdi            / Save arg1
        push    %rsi            / Save arg2
        push    %rdx            / Save arg3
        call    arcfour_crypt_on_intel
        pop     %rdx            / Restore arg3
        pop     %rsi            / Restore arg2
        pop     %rdi            / Restore arg1
        / Save return value in key->flag (1=Intel, 0=AMD)
        movl    %eax,1032($dat)

        / Set $dat to beginning of array, key->arr[0]
        lea     8($dat),$dat
        lea     ($inp,$len),$inp
        neg     $len
        mov     $len,%rcx

        xor     %eax,%eax
        xor     $ido,$ido
        xor     %r10,%r10
        xor     %r11,%r11

        / Use a 4-byte data array
        jmp     .Lw1stloop

.align  16
.Lw1stloop:
        / AMD64 (4-byte array)
        mov     %eax,($dat,%rax,4)
        add     \$1,%al
        jnc     .Lw1stloop

        xor     $ido,$ido
        xor     $idx,$idx

.align  16
.Lw2ndloop:
        mov     ($dat,$ido,4),%r10d
        add     ($inp,$len,1),$idx#b
        add     %r10b,$idx#b
        add     \$1,$len
        mov     ($dat,$idx,4),%r11d
        cmovz   %rcx,$len
        mov     %r10d,($dat,$idx,4)
        mov     %r11d,($dat,$ido,4)
        add     \$1,$ido#b
```



```
        jnc     .Lw2ndloop

        / Exit code
        xor     %eax,%eax
        mov     %eax,-8($dat)
        mov     %eax,-4($dat)

        /* EXPORT DELETE END */
        ret
SET_SIZE(arcfour_key_init)
```

unchanged_portion_omitted_

```
**********************************************************
    5307 Thu Jul 11 01:29:02 2013
new/usr/src/common/crypto/arcfour/arcfour_crypt.c
first pass
**********************************************************
```

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

#define ARCFOUR_LOOP_OPTIMIZED

#ifndef _KERNEL
#include <stdint.h>
#endif  /* _KERNEL */

#include "arcfour.h"

#if defined(__amd64)
/* ARCFour_key.flag values */
#define ARCFOUR_ON_INTEL        1
#define ARCFOUR_ON_AMD64        0

#ifdef _KERNEL
#include <sys/x86_archext.h>
#include <sys/cpuvar.h>

#else
#include <sys/auxv.h>
#endif  /* _KERNEL */
#endif  /* __amd64 */

#ifndef __amd64
/*
 * Initialize the key stream 'key' using the key value.
 *
 * Input:
 * keyval       User-provided key
 * keyvallen    Length, in bytes, of keyval
 * Output:
 * key          Initialized ARCFOUR key schedule, based on keyval
 */
void
arcfour_key_init(ARCFour_key *key, uchar_t *keyval, int keyvallen)
{
/* EXPORT DELETE START */
```



```
        uchar_t ext_keyval[256];
        uchar_t tmp;
        int i, j;

        /* Normalize key length to 256 */
        for (i = j = 0; i < 256; i++, j++) {
                if (j == keyvallen)
                        j = 0;
                ext_keyval[i] = keyval[j];
        }

        for (i = 0; i < 256; i++)
                key->arr[i] = (uchar_t)i;

        j = 0;
        for (i = 0; i < 256; i++) {
                j = (j + key->arr[i] + ext_keyval[i]) & 0xff;
                tmp = key->arr[i];
                key->arr[i] = key->arr[j];
                key->arr[j] = tmp;
        }
        key->i = 0;
        key->j = 0;

/* EXPORT DELETE END */
}
#endif  /* !__amd64 */


/*
 * Encipher 'in' using 'key'.
 *
 * Input:
 * key          ARCFOUR key, initialized by arcfour_key_init()
 * in           Input text
 * out          Buffer to contain output text
 * len          Length, in bytes, of the in and out buffers
 *
 * Output:
 * out          Buffer containing output text
 *
 * Note: in and out can point to the same location
 */
void
arcfour_crypt(ARCFour_key *key, uchar_t *in, uchar_t *out, size_t len)
{
/* EXPORT DELETE START */
#ifdef  __amd64
        if (key->flag == ARCFOUR_ON_AMD64) {
                arcfour_crypt_asm(key, in, out, len);
        } else { /* Intel EM64T */
#endif  /* amd64 */

        size_t          ii;
        uchar_t         i, j, ti, tj;
#ifdef ARCFOUR_LOOP_OPTIMIZED
        uchar_t         arr_ij;
#endif
#ifdef __amd64
        uint32_t        *arr;
#else
        uchar_t         *arr;
#endif

#ifdef  sun4u
```

```
123         /*
124          * The sun4u has a version of arcfour_crypt_aligned() hand-tuned for
125          * the cases where the input and output buffers are aligned on
126          * a multiple of 8-byte boundary.
127          */
128         int             index;
129         uchar_t         tmp;

131         index = (((uint64_t)(uintptr_t)in) & 0x7);

133         /* Get the 'in' on an 8-byte alignment */
134         if (index > 0) {
135                 i = key->i;
136                 j = key->j;
137                 for (index = 8 - (uint64_t)(uintptr_t)in & 0x7;
138                     (index-- > 0) && len > 0;
139                     len--, in++, out++) {
140                         ++i;
141                         j = j + key->arr[i];
142                         tmp = key->arr[i];
143                         key->arr[i] = key->arr[j];
144                         key->arr[j] = tmp;
145                         tmp = key->arr[i] + key->arr[j];
146                         *out = *in ^ key->arr[tmp];
147                 }
148                 key->i = i;
149                 key->j = j;
150         }

152         if (len == 0)
153                 return;

155         /* See if we're fortunate and 'out' got aligned as well */

157         if ((((uint64_t)(uintptr_t)out) & 7) != 0) {
158 #endif  /* sun4u */

160         i = key->i;
161         j = key->j;
162         arr = key->arr;

164 #ifndef ARCFOUR_LOOP_OPTIMIZED
165         /*
166          * This loop is hasn't been reordered, but is kept for reference
167          * purposes as it's more readable
168          */
169         for (ii = 0; ii < len; ++ii) {
170                 ++i;
171                 ti = arr[i];
172                 j = j + ti;
173                 tj = arr[j];
174                 arr[j] = ti;
175                 arr[i] = tj;
176                 out[ii] = in[ii] ^ arr[(ti + tj) & 0xff];
177         }

179 #else
180         /*
181          * This for loop is optimized by carefully spreading out
182          * memory access and storage to avoid conflicts,
183          * allowing the processor to process operations in parallel
184          */

186         /* for loop setup */
187         ++i;
188         ti = arr[i];
```



```
189         j = j + ti;
190         tj = arr[j];
191         arr[j] = ti;
192         arr[i] = tj;
193         arr_ij = arr[(ti + tj) & 0xff];
194         --len;

196         for (ii = 0; ii < len; ) {
197                 ++i;
198                 ti = arr[i];
199                 j = j + ti;
200                 tj = arr[j];
201                 arr[j] = ti;
202                 arr[i] = tj;

204                 /* save result from previous loop: */
205                 out[ii] = in[ii] ^ arr_ij;

207                 ++ii;
208                 arr_ij = arr[(ti + tj) & 0xff];
209         }
210         /* save result from last loop: */
211         out[ii] = in[ii] ^ arr_ij;
212 #endif

214         key->i = i;
215         key->j = j;

217 #ifdef  sun4u
218         } else {
219                 arcfour_crypt_aligned(key, len, in, out);
220         }
221 #endif  /* sun4u */
222 #ifdef  __amd64
223         }
224 #endif  /* amd64 */

231 /* EXPORT DELETE END */
225 }
```

_____unchanged_portion_omitted_

```
**********************************************************
    8003 Thu Jul 11 01:29:03 2013
new/usr/src/common/crypto/arcfour/sun4u/arcfour_crypt_asm.s
first pass
**********************************************************
```

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

#if defined(lint) || defined(__lint)

#include "arcfour.h"

/* ARGSUSED */
void
arcfour_crypt_aligned(ARCFour_key *key, size_t len, uchar_t *in, uchar_t *out)
{}

#else   /* lint || __lint */

        .register       %g2,#scratch
        .register       %g3,#scratch

        .section        ".text",#alloc,#execinstr
        .file   "arcfour_crypt_asm.s"

        .section        ".text",#alloc
        .align  32

        .section        ".text",#alloc,#execinstr
        .align  32
        .skip   32

/*
 * SUBROUTINE arcfour_crypt_aligned
 *
 * void arcfour_crypt_aligned(ARCFour_key *key, size_t len,
 *                      uchar_t *in, uchar_t *out);
 *
 * in and out should be aligned on an 8-byte boundary, but len can be anything
 */
        .global arcfour_crypt_aligned


arcfour_crypt_aligned:
```



```
/* EXPORT DELETE START */
        save    %sp,-144,%sp

        srl     %i1, 3, %l7
        ldub    [%i0+256], %g1

        orcc    %l7, %g0, %g0
        ldub    [%i0+257], %g2

        add     %g1, 1, %o1
        bz      %icc, .Loop2
        add     %i0, 0, %i5

        add     %o1, 1, %g1
        and     %o1, 255, %o1

        and     %g1, 255, %g1
        ldub    [%i5 + %o1], %o3

        ldub    [%i5 + %g1], %g3

        add     %g2, %o3, %o2

        add     %o2, %g3, %g2
        and     %o2, 255, %o2

        and     %g2, 255, %g2
        ldub    [%i5 + %o2], %o4

        stb     %o3, [%i5+%o2]
        subcc   %o2, %g1, %g0

        stb     %o4, [%i5 + %o1]
        bz      %icc, .L1A
        add     %o3,%o4,%o5
.L1B:
        and     %o5, 255, %o5
        ldub    [%i5 + %g2], %g4

        ldub    [%i5 + %o5], %o5
        add     %g1, 1, %o1

        and     %o1, 255, %o1
        stb     %g3, [%i5 + %g2]
        add     %g3, %g4, %g5

        and     %g5, 255, %g5
        stb     %g4, [%i5 + %g1]
        add     %o1, 1, %g1


        sllx    %o5, 56, %o0
        ldub    [%i5 + %o1], %o3
        and     %g1, 255, %g1

        ldub    [%i5 + %g1], %g3

        add     %g2, %o3, %o2
        ldub    [%i5 + %g5], %g5

        add     %o2, %g3, %g2
        and     %o2, 255, %o2

        sllx    %g5, 48, %g5
        ldub    [%i5 + %o2], %o4
```

```
126         and     %g2, 255, %g2

128         or      %o0, %g5, %o0
129         stb     %o3, [%i5+%o2]
130         subcc   %o2, %g1, %g0

132         stb     %o4, [%i5 + %o1]
133         bz      %icc, .L2A
134         add     %o3,%o4,%o5
135 .L2B:
136         and     %o5, 255, %o5
137         ldub    [%i5 + %g2], %g4

139         ldub    [%i5 + %o5], %o5
140         add     %g1, 1, %o1

142         and     %o1, 255, %o1
143         stb     %g3, [%i5 + %g2]
144         add     %g3, %g4, %g5

146         and     %g5, 255, %g5
147         stb     %g4, [%i5 + %g1]
148         add     %o1, 1, %g1


151         sllx    %o5, 40, %o5
152         ldub    [%i5 + %o1], %o3
153         and     %g1, 255, %g1

155         ldub    [%i5 + %g1], %g3
156         or      %o0, %o5, %o0

158         add     %g2, %o3, %o2
159         ldub    [%i5 + %g5], %g5

161         add     %o2, %g3, %g2
162         and     %o2, 255, %o2

164         sllx    %g5, 32, %g5
165         ldub    [%i5 + %o2], %o4
166         and     %g2, 255, %g2

168         or      %o0, %g5, %o0
169         stb     %o3, [%i5+%o2]
170         subcc   %o2, %g1, %g0

172         stb     %o4, [%i5 + %o1]
173         bz      %icc, .L3A
174         add     %o3,%o4,%o5
175 .L3B:
176         and     %o5, 255, %o5
177         ldub    [%i5 + %g2], %g4

179         ldub    [%i5 + %o5], %o5
180         add     %g1, 1, %o1

182         and     %o1, 255, %o1
183         stb     %g3, [%i5 + %g2]
184         add     %g3, %g4, %g5

186         and     %g5, 255, %g5
187         stb     %g4, [%i5 + %g1]
188         add     %o1, 1, %g1


191         sll     %o5, 24, %o5
```



```
192         ldub    [%i5 + %o1], %o3
193         and     %g1, 255, %g1

195         sub     %i1, 8, %i1
196         ldub    [%i5 + %g1], %g3
197         or      %o0, %o5, %o0

199         srl     %i1, 3, %l7
200         ldub    [%i5 + %g5], %g5
201         add     %g2, %o3, %o2

203         add     %o2, %g3, %g2
204         and     %o2, 255, %o2

206         sll     %g5, 16, %g5
207         ldub    [%i5 + %o2], %o4
208         and     %g2, 255, %g2

210         or      %o0, %g5, %o0
211         stb     %o3, [%i5+%o2]
212         subcc   %o2, %g1, %g0

214         stb     %o4, [%i5 + %o1]
215         bz      %icc, .L4A
216         add     %o3,%o4,%o5
217 .L4B:
218         and     %o5, 255, %o5
219         ldub    [%i5 + %g2], %g4
220         add     %g1, 1, %o1

222         orcc    %l7, %g0, %g0
223         ldub    [%i5 + %o5], %o5
224         and     %o1, 255, %o1

226         add     %g3, %g4, %g5
227         stb     %g4, [%i5 + %g1]
228         add     %o1, 1, %g1

230         stb     %g3, [%i5 + %g2]
231         bz      %icc, .EndLoop1
232         and     %g5, 255, %g5


235 .Loop1:
236         sll     %o5, 8, %o5
237         ldub    [%i5 + %o1], %o3
238         and     %g1, 255, %g1

240         ldub    [%i5 + %g1], %g3
241         or      %o0, %o5, %o0

243         ldub    [%i5 + %g5], %g5
244         add     %g2, %o3, %o2

246         add     %o2, %g3, %g2
247         ldx     [%i2], %o7
248         and     %o2, 255, %o2

250         and     %g2, 255, %g2
251         ldub    [%i5 + %o2], %o4

253         or      %o0, %g5, %o0
254         stb     %o3, [%i5+%o2]
255         subcc   %o2, %g1, %g0

257         stb     %o4, [%i5 + %o1]
```

```
258         bz      %icc, .L5A
259         add     %o3,%o4,%o5
260 .L5B:
261         and     %o5, 255, %o5
262         ldub    [%i5 + %g2], %g4

264         ldub    [%i5 + %o5], %o5
265         add     %g1, 1, %o1

267         and     %o1, 255, %o1
268         stb     %g3, [%i5 + %g2]
269         add     %g3, %g4, %g5

271         and     %g5, 255, %g5
272         stb     %g4, [%i5 + %g1]
273         add     %o1, 1, %g1


276         xor     %o0, %o7, %o7
277         ldub    [%i5 + %o1], %o3
278         and     %g1, 255, %g1

280         sllx    %o5, 56, %o0
281         ldub    [%i5 + %g1], %g3

283         add     %g2, %o3, %o2
284         ldub    [%i5 + %g5], %g5

286         add     %o2, %g3, %g2
287         stx     %o7, [%i3]
288         and     %o2, 255, %o2

290         sllx    %g5, 48, %g5
291         ldub    [%i5 + %o2], %o4
292         and     %g2, 255, %g2

294         or      %o0, %g5, %o0
295         stb     %o3, [%i5+%o2]
296         subcc   %o2, %g1, %g0

298         stb     %o4, [%i5 + %o1]
299         bz      %icc, .L6A
300         add     %o3,%o4,%o5
301 .L6B:
302         and     %o5, 255, %o5
303         ldub    [%i5 + %g2], %g4
304         add     %i3, 8, %i3

306         add     %i2, 8, %i2
307         ldub    [%i5 + %o5], %o5
308         add     %g1, 1, %o1

310         and     %o1, 255, %o1
311         stb     %g3, [%i5 + %g2]
312         add     %g3, %g4, %g5

314         and     %g5, 255, %g5
315         stb     %g4, [%i5 + %g1]
316         add     %o1, 1, %g1


319         sllx    %o5, 40, %o5
320         ldub    [%i5 + %o1], %o3
321         and     %g1, 255, %g1

323         ldub    [%i5 + %g1], %g3
```



```
324         or      %o0, %o5, %o0

326         add     %g2, %o3, %o2
327         ldub    [%i5 + %g5], %g5

329         add     %o2, %g3, %g2
330         and     %o2, 255, %o2

332         sllx    %g5, 32, %g5
333         ldub    [%i5 + %o2], %o4
334         and     %g2, 255, %g2

336         or      %o0, %g5, %o0
337         stb     %o3, [%i5 + %o2]
338         subcc   %o2, %g1, %g0

340         stb     %o4, [%i5 + %o1]
341         bz      %icc, .L7A
342         add     %o3,%o4,%o5
343 .L7B:
344         and     %o5, 255, %o5
345         ldub    [%i5 + %g2], %g4

347         ldub    [%i5 + %o5], %o5
348         add     %g1, 1, %o1

350         and     %o1, 255, %o1
351         stb     %g3, [%i5 + %g2]
352         add     %g3, %g4, %g5

354         and     %g5, 255, %g5
355         stb     %g4, [%i5 + %g1]
356         add     %o1, 1, %g1


359         sll     %o5, 24, %o5
360         ldub    [%i5 + %o1], %o3
361         and     %g1, 255, %g1

363         sub     %i1, 8, %i1
364         ldub    [%i5 + %g1], %g3
365         or      %o0, %o5, %o0

367         srl     %i1, 3, %l7
368         ldub    [%i5 + %g5], %g5
369         add     %g2, %o3, %o2

371         add     %o2, %g3, %g2
372         and     %o2, 255, %o2

374         sll     %g5, 16, %g5
375         ldub    [%i5 + %o2], %o4
376         and     %g2, 255, %g2

378         or      %o0, %g5, %o0
379         stb     %o3, [%i5 + %o2]
380         subcc   %o2, %g1, %g0

382         stb     %o4, [%i5 + %o1]
383         bz      %icc, .L8A
384         add     %o3,%o4,%o5
385 .L8B:
386         and     %o5, 255, %o5
387         ldub    [%i5 + %g2], %g4
388         add     %g1, 1, %o1
```

```
390         orcc    %l7, %g0, %g0
391         ldub    [%i5 + %o5], %o5
392         and     %o1, 255, %o1

394         add     %g3, %g4, %g5
395         stb     %g4, [%i5 + %g1]
396         add     %o1, 1, %g1

398         stb     %g3, [%i5 + %g2]
399         bnz     %icc, .Loop1
400         and     %g5, 255, %g5


403 .EndLoop1:
404         sll     %o5, 8, %o5
405         ldub    [%i5 + %g5], %g5
406         orcc    %i1, %g0, %g0

408         or      %o0, %o5, %o0
409         ldx     [%i2], %o7
410         sub     %g1, 2, %g1

412         and     %g1, 255, %g1
413         stb     %g1, [%i0 + 256]
414         or      %o0, %g5, %o0

416         xor     %o0, %o7, %o7
417         stx     %o7, [%i3]
418         add     %i2, 8, %i2

420         add     %i3, 8, %i3
421         bnz     %icc, .Loop2_1
422         stb     %g2, [%i0 + 257]

424         ret
425         restore %g0,%g0,%g0


428 .Loop2:
429         orcc    %i1, %g0, %g0
430         bnz     .Loop2_1
431         nop
432         ret
433         restore %g0,%g0,%g0

435 .Loop2_1:
436         and     %o1, 255, %g1
437         ldub    [%i5 + %g1], %g3

439         add     %g2, %g3, %g2

441         and     %g2, 255, %g2

443         ldub    [%i5 + %g2], %g4

445         stb     %g3, [%i5 + %g2]

447         add     %g3, %g4, %g5
448         stb     %g4, [%i5 + %g1]

450         and     %g5, 255, %g5
451         ldub    [%i2], %o0

453         add     %g1, 1, %o1
454         ldub    [%i5 + %g5], %g5
455         subcc   %i1, 1, %i1
```



```
457         add     %i2, 1, %i2
458         add     %i3, 1, %i3

460         xor     %o0, %g5, %o0
461         bnz     %icc, .Loop2_1
462         stb     %o0, [%i3 - 1]

464         stb     %g1, [%i0 + 256]

466         stb     %g2, [%i0 + 257]

468         ret
469         restore %g0,%g0,%g0

471 .L1A:
472         add     %o2, %o3, %g2
473         or      %o3, %g0, %g3
474         ba      .L1B
475         and     %g2, 255, %g2

477 .L2A:
478         add     %o2, %o3, %g2
479         or      %o3, %g0, %g3
480         ba      .L2B
481         and     %g2, 255, %g2

483 .L3A:
484         add     %o2, %o3, %g2
485         or      %o3, %g0, %g3
486         ba      .L3B
487         and     %g2, 255, %g2

489 .L4A:
490         add     %o2, %o3, %g2
491         or      %o3, %g0, %g3
492         ba      .L4B
493         and     %g2, 255, %g2

495 .L5A:
496         add     %o2, %o3, %g2
497         or      %o3, %g0, %g3
498         ba      .L5B
499         and     %g2, 255, %g2

501 .L6A:
502         add     %o2, %o3, %g2
503         or      %o3, %g0, %g3
504         ba      .L6B
505         and     %g2, 255, %g2

507 .L7A:
508         add     %o2, %o3, %g2
509         or      %o3, %g0, %g3
510         ba      .L7B
511         and     %g2, 255, %g2

513 .L8A:
514         add     %o2, %o3, %g2
515         or      %o3, %g0, %g3
516         ba      .L8B
517         and     %g2, 255, %g2

521 /* EXPORT DELETE END */
519         .type   arcfour_crypt_aligned,2
520         .size   arcfour_crypt_aligned,(. - arcfour_crypt_aligned)
```

```
 522 #endif  /* lint || __lint */
```

```
**********************************************************
   11847 Thu Jul 11 01:29:04 2013
new/usr/src/common/crypto/arcfour/sun4v/arcfour_crypt.c
first pass
**********************************************************
```

```
/*
 * 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) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
 */

#include "../arcfour.h"

/* Initialize the key stream 'key' using the key value */
void
arcfour_key_init(ARCFour_key *key, uchar_t *keyval, int keyvallen)
{
/* EXPORT DELETE START */

        uchar_t ext_keyval[256];
        uchar_t tmp;
        int i, j;

        for (i = j = 0; i < 256; i++, j++) {
                if (j == keyvallen)
                        j = 0;

                ext_keyval[i] = keyval[j];
        }
        for (i = 0; i < 256; i++)
                key->arr[i] = (uchar_t)i;

        j = 0;
        for (i = 0; i < 256; i++) {
                j = (j + key->arr[i] + ext_keyval[i]) % 256;
                tmp = key->arr[i];
                key->arr[i] = key->arr[j];
                key->arr[j] = tmp;
        }
        key->i = 0;
        key->j = 0;

/* EXPORT DELETE END */
}


/*
```



```
 * Encipher 'in' using 'key.
 * in and out can point to the same location
 */
void
arcfour_crypt(ARCFour_key *key, uchar_t *in, uchar_t *out, size_t len)
{
        size_t ii;
        unsigned long long in0, merge = 0, merge0 = 0, merge1, mask = 0;
        uchar_t i, j, *base, jj, *base1, tmp;
        unsigned int tmp0, tmp1, i_accum, shift = 0, i1;

/* EXPORT DELETE START */
        int index;

        base = key->arr;

        index = (((uintptr_t)in) & 0x7);

        /* Get the 'in' on an 8-byte alignment */
        if (index > 0) {
                i = key->i;
                j = key->j;

                for (index = 8 - index; (index-- > 0) && len > 0;
                    len--, in++, out++) {

                        i = i + 1;
                        j = j + key->arr[i];
                        tmp = key->arr[i];
                        key->arr[i] = key->arr[j];
                        key->arr[j] = tmp;
                        tmp = key->arr[i] + key->arr[j];
                        *out = *in ^ key->arr[tmp];
                }
                key->i = i;
                key->j = j;

        }
        if (len == 0)
                return;

        /* See if we're fortunate and 'out' got aligned as well */

        /*
         * Niagara optimized version for
         * the cases where the input and output  buffers are aligned on
         * a multiple of 8-byte boundary.
         */
#ifdef  sun4v
        if ((((uintptr_t)out) & 7) != 0) {
#endif  /* sun4v */
                i = key->i;
                j = key->j;
                for (ii = 0; ii < len; ii++) {
                        i = i + 1;
                        tmp0 = base[i];
                        j = j + tmp0;
                        tmp1 = base[j];
                        base[i] = (uchar_t)tmp1;
                        base[j] = (uchar_t)tmp0;
                        tmp0 += tmp1;
                        tmp0 = tmp0 & 0xff;
                        out[ii] = in[ii] ^ base[tmp0];
                }
```

```
122                 key->i = i;
123                 key->j = j;
124 #ifdef  sun4v
125         } else {
126                 i = key->i;
127                 j = key->j;

129                 /*
130                  * Want to align base[i] on a 2B boundary -- allows updates
131                  * via [i] to be performed in 2B chunks (reducing # of stores).
132                  * Requires appropriate alias detection.
133                  */

135                 if (((i+1) % 2) != 0) {
136                         i = i + 1;
137                         tmp0 = base[i];
138                         j = j + tmp0;
139                         tmp1 = base[j];

141                         base[i] = (uchar_t)tmp1;
142                         base[j] = (uchar_t)tmp0;

144                         tmp0 += tmp1;
145                         tmp0 = tmp0 & 0xff;

147                         merge0 = (unsigned long long)(base[tmp0]) << 56;
148                         shift = 8; mask = 0xff;
149                 }

151                 /*
152                  * Note - in and out may now be misaligned -
153                  * as updating [out] in 8B chunks need to handle this
154                  * possibility. Also could have a 1B overrun.
155                  * Need to drop out of loop early as a result.
156                  */

158                 for (ii = 0, i1 = i; ii < ((len-1)  & (~7));
159                     ii += 8, i1 = i1&0xff) {

161                         /*
162                          * If i < less than 248, know wont wrap around
163                          * (i % 256), so don't need to bother with masking i
164                          * after each increment
165                          */
166                         if (i1 < 248) {

168                                 /* BYTE 0 */
169                                 i1 = (i1 + 1);

171                                 /*
172                                  * Creating this base pointer reduces subsequent
173                                  * arihmetic ops required to load [i]
174                                  *
175                                  * N.B. don't need to check if [j] aliases.
176                                  * [i] and [j] end up with the same values
177                                  * anyway.
178                                  */
179                                 base1 = &base[i1];

181                                 tmp0 = base1[0];
182                                 j = j + tmp0;

184                                 tmp1 = base[j];
185                                 /*
186                                  * Don't store [i] yet
187                                  */
```



```
188                                 i_accum = tmp1;
189                                 base[j] = (uchar_t)tmp0;

191                                 tmp0 += tmp1;
192                                 tmp0 = tmp0 & 0xff;

194                                 /*
195                                  * Check [tmp0] doesn't alias with [i]
196                                  */

198                                 /*
199                                  * Updating [out] in 8B chunks
200                                  */
201                                 if (i1 == tmp0) {
202                                         merge =
203                                             (unsigned long long)(i_accum) << 56;
204                                 } else {
205                                         merge =
206                                             (unsigned long long)(base[tmp0]) <<
207                                             56;
208                                 }

210                                 /* BYTE 1 */
211                                 tmp0 = base1[1];

213                                 j = j + tmp0;

215                                 /*
216                                  * [j] can now alias with [i] and [i-1]
217                                  * If alias abort speculation
218                                  */
219                                 if ((i1 ^ j) < 2) {
220                                         base1[0] = (uchar_t)i_accum;

222                                         tmp1 = base[j];

224                                         base1[1] = (uchar_t)tmp1;
225                                         base[j] = (uchar_t)tmp0;

227                                         tmp0 += tmp1;
228                                         tmp0 = tmp0 & 0xff;

230                                         merge |= (unsigned long long)
231                                             (base[tmp0]) << 48;
232                                 } else {

234                                         tmp1 = base[j];

236                                         i_accum = i_accum << 8;
237                                         i_accum |= tmp1;

239                                         base[j] = (uchar_t)tmp0;

241                                         tmp0 += tmp1;
242                                         tmp0 = tmp0 & 0xff;

244                                         /*
245                                          * Speculation suceeded! Update [i]
246                                          * in 2B chunk
247                                          */
248                                         /* LINTED E_BAD_PTR_CAST_ALIGN */
249                                         *((unsigned short *) &base[i1]) =
250                                             i_accum;

252                                         merge |=
253                                             (unsigned long long)(base[tmp0]) <<
```

```
254                                             48;
255                             }


258                             /*
259                              * Too expensive to perform [i] speculation for
260                              * every byte. Just need to reduce frequency
261                              * of stores until store buffer full stalls
262                              * are not the bottleneck.
263                              */

265                             /* BYTE 2 */
266                             tmp0 = base1[2];
267                             j = j + tmp0;
268                             tmp1 = base[j];
269                             base1[2] = (uchar_t)tmp1;
270                             base[j] = (uchar_t)tmp0;
271                             tmp1 += tmp0;
272                             tmp1 = tmp1 & 0xff;
273                             merge |= (unsigned long long)(base[tmp1]) << 40;

275                             /* BYTE 3 */
276                             tmp0 = base1[3];
277                             j = j + tmp0;
278                             tmp1 = base[j];
279                             base1[3] = (uchar_t)tmp1;
280                             base[j] = (uchar_t)tmp0;
281                             tmp0 += tmp1;
282                             tmp0 = tmp0 & 0xff;
283                             merge |= (unsigned long long)(base[tmp0]) << 32;

285                             /* BYTE 4 */
286                             tmp0 = base1[4];
287                             j = j + tmp0;
288                             tmp1 = base[j];
289                             base1[4] = (uchar_t)tmp1;
290                             base[j] = (uchar_t)tmp0;
291                             tmp0 += tmp1;
292                             tmp0 = tmp0 & 0xff;
293                             merge |= (unsigned long long)(base[tmp0]) << 24;

295                             /* BYTE 5 */
296                             tmp0 = base1[5];
297                             j = j + tmp0;
298                             tmp1 = base[j];
299                             base1[5] = (uchar_t)tmp1;
300                             base[j] = (uchar_t)tmp0;
301                             tmp0 += tmp1;
302                             tmp0 = tmp0 & 0xff;
303                             merge |= (unsigned long long)(base[tmp0]) << 16;

305                             /* BYTE 6 */
306                             i1 = (i1+6);
307                             tmp0 = base1[6];
308                             j = j + tmp0;
309                             tmp1 = base[j];
310                             i_accum = tmp1;
311                             base[j] = (uchar_t)tmp0;

313                             tmp0 += tmp1;
314                             tmp0 = tmp0 & 0xff;

316                             if (i1 == tmp0) {
317                                     merge |=
318                                         (unsigned long long)(i_accum) << 8;
319                             } else {
```



```
320                                     merge |=
321                                         (unsigned long long)(base[tmp0]) <<
322                                         8;
323                             }

325                             /* BYTE 7 */
326                             tmp0 = base1[7];

328                             /*
329                              * Perform [i] speculation again. Indentical
330                              * to that performed for BYTE0 and BYTE1.
331                              */
332                             j = j + tmp0;
333                             if ((i1 ^ j) < 2) {
334                                     base1[6] = (uchar_t)i_accum;
335                                     tmp1 = base[j];

337                                     base1[7] = (uchar_t)tmp1;
338                                     base[j] = (uchar_t)tmp0;

340                                     tmp0 += tmp1;
341                                     tmp0 = tmp0 & 0xff;

343                                     merge |=
344                                         (unsigned long long)(base[tmp0]);

346                             } else {
347                                     tmp1 = base[j];

349                                     i_accum = i_accum << 8;
350                                     i_accum |= tmp1;

352                                     base[j] = (uchar_t)tmp0;

354                                     tmp0 += tmp1;
355                                     tmp0 = tmp0 & 0xff;

357                                     /* LINTED E_BAD_PTR_CAST_ALIGN */
358                                     *((unsigned short *) &base[i1]) =
359                                         i_accum;

361                                     merge |=
362                                         (unsigned long long)(base[tmp0]);
363                             }
364                             i1++;
365                     } else {
366                             /*
367                              * i is too close to wrap-around to allow
368                              * masking to be disregarded
369                              */

371                             /*
372                              * Same old speculation for BYTE 0 and BYTE 1
373                              */

375                             /* BYTE 0 */
376                             i1 = (i1 + 1) & 0xff;
377                             jj = (uchar_t)i1;

379                             tmp0 = base[i1];
380                             j = j + tmp0;

382                             tmp1 = base[j];
383                             i_accum = tmp1;
384                             base[j] = (uchar_t)tmp0;
```

```
386                             tmp0 += tmp1;
387                             tmp0 = tmp0 & 0xff;

389                             if (i1 == tmp0) {
390                                     merge =
391                                         (unsigned long long)(i_accum) << 56;
392                             } else {
393                                     merge =
394                                         (unsigned long long)(base[tmp0]) <<
395                                         56;
396                             }

398                             /* BYTE 1 */
399                             tmp0 = base[i1+1];

401                             j = j + tmp0;

403                             if ((jj ^ j) < 2) {
404                                     base[jj] = (uchar_t)i_accum;

406                                     tmp1 = base[j];

408                                     base[i1+1] = (uchar_t)tmp1;
409                                     base[j] = (uchar_t)tmp0;

411                                     tmp0 += tmp1;
412                                     tmp0 = tmp0 & 0xff;

414                                     merge |=
415                                         (unsigned long long)(base[tmp0]) <<
416                                         48;
417                             } else {

419                                     tmp1 = base[j];

421                                     i_accum = i_accum << 8;
422                                     i_accum |= tmp1;

424                                     base[j] = (uchar_t)tmp0;

426                                     tmp0 += tmp1;
427                                     tmp0 = tmp0 & 0xff;

429                                     /* LINTED E_BAD_PTR_CAST_ALIGN */
430                                     *((unsigned short *) &base[jj]) =
431                                         i_accum;

433                                     merge |=
434                                         (unsigned long long)(base[tmp0]) <<
435                                         48;
436                             }

438                             /* BYTE 2 */
439                             /*
440                              * As know i must be even when enter loop (to
441                              * satisfy alignment), can only wrap around
442                              * on the even bytes. So just need to perform
443                              * mask every 2nd byte
444                              */
445                             i1 = (i1 + 2) & 0xff;
446                             tmp0 = base[i1];
447                             j = j + tmp0;
448                             tmp1 = base[j];
449                             base[i1] = (uchar_t)tmp1;
450                             base[j] = (uchar_t)tmp0;
451                             tmp0 += tmp1;
```



```
452                             tmp0 = tmp0 & 0xff;
453                             merge |= (unsigned long long)(base[tmp0]) << 40;

455                             /* BYTE 3 */
456                             tmp0 = base[i1+1];
457                             j = j + tmp0;
458                             tmp1 = base[j];
459                             base[i1+1] = (uchar_t)tmp1;
460                             base[j] = (uchar_t)tmp0;
461                             tmp0 += tmp1;
462                             tmp0 = tmp0 & 0xff;
463                             merge |= (unsigned long long)(base[tmp0]) << 32;

465                             /* BYTE 4 */
466                             i1 = (i1 + 2) & 0xff;
467                             tmp0 = base[i1];
468                             j = j + tmp0;
469                             tmp1 = base[j];
470                             base[i1] = (uchar_t)tmp1;
471                             base[j] = (uchar_t)tmp0;
472                             tmp0 += tmp1;
473                             tmp0 = tmp0 & 0xff;
474                             merge |= (unsigned long long)(base[tmp0]) << 24;

476                             /* BYTE 5 */
477                             tmp0 = base[i1+1];
478                             j = j + tmp0;
479                             tmp1 = base[j];
480                             base[i1+1] = (uchar_t)tmp1;
481                             base[j] = (uchar_t)tmp0;
482                             tmp0 += tmp1;
483                             tmp0 = tmp0 & 0xff;
484                             merge |= (unsigned long long)(base[tmp0]) << 16;

486                             /* BYTE 6 */
487                             i1 = (i1+2) &0xff;
488                             jj = (uchar_t)i1;
489                             tmp0 = base[i1];

491                             j = j + tmp0;

493                             tmp1 = base[j];
494                             i_accum = tmp1;
495                             base[j] = (uchar_t)tmp0;


498                             tmp0 += tmp1;
499                             tmp0 = tmp0 & 0xff;

501                             if (i1 == tmp0) {
502                                     merge |=
503                                         (unsigned long long)(i_accum) << 8;
504                             } else {
505                                     merge |=
506                                         (unsigned long long)(base[tmp0]) <<
507                                         8;
508                             }

510                             /* BYTE 7 */
511                             i1++;
512                             tmp0 = base[i1];

514                             j = j + tmp0;
515                             if ((jj ^ j) < 2) {
516                                     base[jj] = (uchar_t)i_accum;
517                                     tmp1 = base[j];
```

```
519                                     base[i1] = (uchar_t)tmp1;
520                                     base[j] = (uchar_t)tmp0;

522                                     tmp0 += tmp1;
523                                     tmp0 = tmp0 & 0xff;

525                                     merge |=
526                                         (unsigned long long)(base[tmp0]);

528                             } else {

530                                     tmp1 = base[j];

532                                     i_accum = i_accum << 8;
533                                     i_accum |= tmp1;

535                                     base[j] = (uchar_t)tmp0;

537                                     tmp0 += tmp1;
538                                     tmp0 = tmp0 & 0xff;

540                                     /* LINTED E_BAD_PTR_CAST_ALIGN */
541                                     *((unsigned short *) &base[jj]) =
542                                         i_accum;

544                                     merge |=
545                                         (unsigned long long)(base[tmp0]);
546                             }
547                     }

549                     /*
550                      * Perform update to [out]
551                      * Remember could be alignment issues
552                      */
553                     /* LINTED E_BAD_PTR_CAST_ALIGN */
554                     in0 = *((unsigned long long *) (&in[ii]));

556                     merge1 = merge0 | (merge >> shift);

558                     merge0 = (merge & mask) << 56;

560                     in0 = in0 ^ merge1;

562                     /* LINTED E_BAD_PTR_CAST_ALIGN */
563                     *((unsigned long long *) (&out[ii])) = in0;
564             }

566             i = (uchar_t)i1;

568             /*
569              * Handle any overrun
570              */
571             if (shift) {
572                     out[ii] = in[ii] ^ (merge0 >> 56);
573                     ii++;
574             }

576             /*
577              * Handle final few bytes
578              */
579             for (; ii < len; ii++) {
580                     i = i + 1;
581                     tmp0 = base[i];
582                     j = j + tmp0;
583                     tmp1 = base[j];
```



```
585                             base[i] = (uchar_t)tmp1;
586                             base[j] = (uchar_t)tmp0;

588                             tmp0 += tmp1;
589                             tmp0 = tmp0 & 0xff;
590                             out[ii] = in[ii] ^ base[tmp0];
591                     }
592                     key->i = i;
593                     key->j = j;
594             }
595 #endif /* sun4v */

603 /* EXPORT DELETE END */
596 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1045 Thu Jul 11 01:29:04 2013
new/usr/src/common/crypto/blowfish/Makefile
first pass
**********************************************************
```

```
#
# 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 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# ident "%Z%%M% %I%     %E% SMI"
#
# common/crypto/blowfish/Makefile
#
# include global definitions
include $(SRC)/Makefile.master

.KEEP_STATE:

FRC:

# EXPORT DELETE START
EXPORT_SRC:
        $(RM) Makefile+ blowfish_impl.c+ blowfish_impl.h+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < blowfish_impl.c > blowfish_impl.c+
        $(MV) blowfish_impl.c+ blowfish_impl.c
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < blowfish_impl.h > blowfish_impl.h+
        $(MV) blowfish_impl.h+ blowfish_impl.h
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(RM) Makefile
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile blowfish_impl.c blowfish_impl.h

# EXPORT DELETE END
```

****
26523 Thu Jul 11 01:29:05 2013
new/usr/src/common/crypto/blowfish/blowfish_impl.c
first pass
****

```
/*
 * 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 2010 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Blowfish encryption/decryption and keyschedule code.
 */

#include <sys/types.h>
#include <sys/systm.h>
#include <sys/ddi.h>
#include <sys/sysmacros.h>
#include <sys/strsun.h>
#include <sys/note.h>
#include <sys/byteorder.h>
#include <sys/crypto/spi.h>
#include <modes/modes.h>
#include <sys/crypto/common.h>
#include "blowfish_impl.h"

#ifdef _KERNEL

#define BLOWFISH_ASSERT(x)      ASSERT(x)

#else /* !_KERNEL */

#include <strings.h>
#include <stdlib.h>
#define BLOWFISH_ASSERT(x)
#endif /* _KERNEL */

#if defined(__i386) || defined(__amd64)
#include <sys/byteorder.h>
#define UNALIGNED_POINTERS_PERMITTED
#endif

/* EXPORT DELETE START */

/*
 * Blowfish initial P box and S boxes, derived from the hex digits of PI.
```



```
 *
 * NOTE:  S boxes are placed into one large array.
 */
static const uint32_t init_P[] = {
        0x243f6a88U, 0x85a308d3U, 0x13198a2eU,
        0x03707344U, 0xa4093822U, 0x299f31d0U,
        0x082efa98U, 0xec4e6c89U, 0x452821e6U,
        0x38d01377U, 0xbe5466cfU, 0x34e90c6cU,
        0xc0ac29b7U, 0xc97c50ddU, 0x3f84d5b5U,
        0xb5470917U, 0x9216d5d9U, 0x8979fb1bU
};
______unchanged_portion_omitted_

/*
 * Since ROUND() is a macro, make sure that the things inside can be
 * evaluated more than once.  Especially when calling F().
 * Assume the presence of local variables:
 *
 *      uint32_t *P;
 *      uint32_t *S;
 *      uint32_t tmp;
 *
 *
 * And to Microsoft interview survivors out there, perhaps I should do the
 * XOR swap trick, or at least #ifdef (__i386) the tmp = ... = tmp; stuff.
 */

#define F(word) \
        (((S[(word >> 24) & 0xff] + S[256 + ((word >> 16) & 0xff)]) ^ \
                S[512 + ((word >> 8) & 0xff)]) + S[768 + (word & 0xff)])

#define ROUND(left, right, i) \
        (left) ^= P[i]; \
        (right) ^= F((left)); \
        tmp = (left); \
        (left) = (right); \
        (right) = tmp;

/* EXPORT DELETE END */

/*
 * Encrypt a block of data.  Because of addition operations, convert blocks
 * to their big-endian representation, even on Intel boxen.
 */
/* ARGSUSED */
int
blowfish_encrypt_block(const void *cookie, const uint8_t *block,
    uint8_t *out_block)
{
/* EXPORT DELETE START */
        keysched_t *ksch = (keysched_t *)cookie;

        uint32_t left, right, tmp;
        uint32_t *P = ksch->ksch_P;
        uint32_t *S = ksch->ksch_S;
#ifdef _BIG_ENDIAN
        uint32_t *b32;

        if (IS_P2ALIGNED(block, sizeof (uint32_t))) {
                /* LINTED:  pointer alignment */
                b32 = (uint32_t *)block;
                left = b32[0];
                right = b32[1];
        } else
#endif
        {
```

```
393         /*
394          * Read input block and place in left/right in big-endian order.
395          */
396 #ifdef UNALIGNED_POINTERS_PERMITTED
397         left = htonl(*(uint32_t *)(void *)&block[0]);
398         right = htonl(*(uint32_t *)(void *)&block[4]);
399 #else
400         left = ((uint32_t)block[0] << 24)
401             | ((uint32_t)block[1] << 16)
402             | ((uint32_t)block[2] << 8)
403             | (uint32_t)block[3];
404         right = ((uint32_t)block[4] << 24)
405             | ((uint32_t)block[5] << 16)
406             | ((uint32_t)block[6] << 8)
407             | (uint32_t)block[7];
408 #endif  /* UNALIGNED_POINTERS_PERMITTED */
409         }

411         ROUND(left, right, 0);
412         ROUND(left, right, 1);
413         ROUND(left, right, 2);
414         ROUND(left, right, 3);
415         ROUND(left, right, 4);
416         ROUND(left, right, 5);
417         ROUND(left, right, 6);
418         ROUND(left, right, 7);
419         ROUND(left, right, 8);
420         ROUND(left, right, 9);
421         ROUND(left, right, 10);
422         ROUND(left, right, 11);
423         ROUND(left, right, 12);
424         ROUND(left, right, 13);
425         ROUND(left, right, 14);
426         ROUND(left, right, 15);

428         tmp = left;
429         left = right;
430         right = tmp;
431         right ^= P[16];
432         left ^= P[17];

434 #ifdef _BIG_ENDIAN
435         if (IS_P2ALIGNED(out_block, sizeof (uint32_t))) {
436                 /* LINTED:  pointer alignment */
437                 b32 = (uint32_t *)out_block;
438                 b32[0] = left;
439                 b32[1] = right;
440         } else
441 #endif
442         {
443                 /* Put the block back into the user's block with final swap */
444 #ifdef UNALIGNED_POINTERS_PERMITTED
445                 *(uint32_t *)(void *)&out_block[0] = htonl(left);
446                 *(uint32_t *)(void *)&out_block[4] = htonl(right);
447 #else
448                 out_block[0] = left >> 24;
449                 out_block[1] = left >> 16;
450                 out_block[2] = left >> 8;
451                 out_block[3] = left;
452                 out_block[4] = right >> 24;
453                 out_block[5] = right >> 16;
454                 out_block[6] = right >> 8;
455                 out_block[7] = right;
456 #endif  /* UNALIGNED_POINTERS_PERMITTED */
457         }
463 /* EXPORT DELETE END */
```



```
458         return (CRYPTO_SUCCESS);
459 }

461 /*
462  * Decrypt a block of data.  Because of addition operations, convert blocks
463  * to their big-endian representation, even on Intel boxen.
464  * It should look like the blowfish_encrypt_block() operation
465  * except for the order in which the S/P boxes are accessed.
466  */
467 /* ARGSUSED */
468 int
469 blowfish_decrypt_block(const void *cookie, const uint8_t *block,
470     uint8_t *out_block)
471 {
478 /* EXPORT DELETE START */
472         keysched_t *ksch = (keysched_t *)cookie;

474         uint32_t left, right, tmp;
475         uint32_t *P = ksch->ksch_P;
476         uint32_t *S = ksch->ksch_S;
477 #ifdef _BIG_ENDIAN
478         uint32_t *b32;

480         if (IS_P2ALIGNED(block, sizeof (uint32_t))) {
481                 /* LINTED:  pointer alignment */
482                 b32 = (uint32_t *)block;
483                 left = b32[0];
484                 right = b32[1];
485         } else
486 #endif
487         {
488         /*
489          * Read input block and place in left/right in big-endian order.
490          */
491 #ifdef UNALIGNED_POINTERS_PERMITTED
492         left = htonl(*(uint32_t *)(void *)&block[0]);
493         right = htonl(*(uint32_t *)(void *)&block[4]);
494 #else
495         left = ((uint32_t)block[0] << 24)
496             | ((uint32_t)block[1] << 16)
497             | ((uint32_t)block[2] << 8)
498             | (uint32_t)block[3];
499         right = ((uint32_t)block[4] << 24)
500             | ((uint32_t)block[5] << 16)
501             | ((uint32_t)block[6] << 8)
502             | (uint32_t)block[7];
503 #endif  /* UNALIGNED_POINTERS_PERMITTED */
504         }

506         ROUND(left, right, 17);
507         ROUND(left, right, 16);
508         ROUND(left, right, 15);
509         ROUND(left, right, 14);
510         ROUND(left, right, 13);
511         ROUND(left, right, 12);
512         ROUND(left, right, 11);
513         ROUND(left, right, 10);
514         ROUND(left, right, 9);
515         ROUND(left, right, 8);
516         ROUND(left, right, 7);
517         ROUND(left, right, 6);
518         ROUND(left, right, 5);
519         ROUND(left, right, 4);
520         ROUND(left, right, 3);
521         ROUND(left, right, 2);
```

```
523         tmp = left;
524         left = right;
525         right = tmp;
526         right ^= P[1];
527         left ^= P[0];

529 #ifdef _BIG_ENDIAN
530         if (IS_P2ALIGNED(out_block, sizeof (uint32_t))) {
531                 /* LINTED:  pointer alignment */
532                 b32 = (uint32_t *)out_block;
533                 b32[0] = left;
534                 b32[1] = right;
535         } else
536 #endif
537         {
538         /* Put the block back into the user's block with final swap */
539 #ifdef UNALIGNED_POINTERS_PERMITTED
540                 *(uint32_t *)(void *)&out_block[0] = htonl(left);
541                 *(uint32_t *)(void *)&out_block[4] = htonl(right);
542 #else
543                 out_block[0] = left >> 24;
544                 out_block[1] = left >> 16;
545                 out_block[2] = left >> 8;
546                 out_block[3] = left;
547                 out_block[4] = right >> 24;
548                 out_block[5] = right >> 16;
549                 out_block[6] = right >> 8;
550                 out_block[7] = right;
551 #endif  /* UNALIGNED_POINTERS_PERMITTED */
552         }
560 /* EXPORT DELETE END */
553         return (CRYPTO_SUCCESS);
554 }

556 static void
557 bitrepeat(uint8_t *pattern, uint_t len_bytes, uint_t len_bits, uint8_t *dst,
558     uint_t dst_len_bytes)
559 {
568 /* EXPORT DELETE START */
560         uint8_t *current = dst;
561         uint_t bitsleft = CRYPTO_BYTES2BITS(dst_len_bytes);
562         uint_t bitoffset = 0;
563         uint_t currentbits;
564         int i;

566         BLOWFISH_ASSERT(CRYPTO_BITS2BYTES(len_bits) == len_bytes);

568         bzero(dst, dst_len_bytes);

570         while (bitsleft != 0) {
571                 if (bitsleft >= len_bits) {
572                         currentbits = len_bits;

574                         for (i = 0; i < len_bytes; i++) {
575                                 if (currentbits >= 8) {
576                                         *current++ |= pattern[i] >> bitoffset;
577                                         *current |= pattern[i] << 8 - bitoffset;
578                                         currentbits -= 8;
579                                 } else {
580                                         *current |= pattern[i] >> bitoffset;
581                                         bitoffset = bitoffset + currentbits;
582                                         bitoffset &= 0x7;
583                                         if (bitoffset == 0)
584                                                 current++;
585                                 }
586                         }
```



```
587                         bitsleft -= len_bits;
588                 } else {
589                         currentbits = bitsleft;

591                         for (i = 0; i < len_bytes && bitsleft != 0; i++) {
592                                 if (currentbits >= 8 &&
593                                     current < dst + dst_len_bytes) {
594                                         *current++ |= pattern[i] >> bitoffset;
595                                         *current |= pattern[i] << 8 - bitoffset;
596                                         currentbits -= 8;
597                                         bitsleft -= 8;
598                                 } else {
599                                         *current |= pattern[i] >> bitoffset;
600                                         bitsleft -= bitoffset;
601                                         bitoffset = bitoffset + currentbits;
602                                         bitoffset &= 0x7;
603                                         if (bitoffset == 0)
604                                                 current++;
605                                         currentbits = 0;
606                                 }
607                         }
608                         bitsleft = 0;
609                 }
610         }
620 /* EXPORT DELETE END */
611 }

613 /*
614  * Initialize key schedules for Blowfish.
615  */
616 void
617 blowfish_init_keysched(uint8_t *key, uint_t bits, void *keysched)
618 {
629 /* EXPORT DELETE START */
619         keysched_t *newbie = keysched;
620         uint32_t *P = newbie->ksch_P;
621         uint32_t *S = newbie->ksch_S;
622         uint32_t *initp;
623         uint32_t tmpblock[] = {0, 0};
624         uint8_t *rawkeybytes = (uint8_t *)P;
625         int i, slop, copylen;
626         uintptr_t bytesleft;
627         uint_t len;

629         len = CRYPTO_BITS2BYTES(bits);

631         if ((bits & 0x7) != 0) {
632                 /*
633                  * Really slow case, bits aren't on a byte boundary.
634                  * Keep track of individual bits copied over.  :-P
635                  */
636                 bitrepeat(key, len, bits, rawkeybytes, 72);
637         } else {
638                 slop = 72 % len;

640                 /* Someone gave us a nice amount (i.e. div by 8) of bits */
641                 while (rawkeybytes != (uint8_t *)(P + 18)) {
642                         bytesleft =
643                             (uintptr_t)(P + 18) - (uintptr_t)rawkeybytes;
644                         copylen = (bytesleft >= len) ? len : slop;
645                         bcopy(key, rawkeybytes, copylen);
646                         rawkeybytes += copylen;
647                 }
648         }

650         for (i = 0; i < 18; i++)
```

```
651                 P[i] = ntohl(P[i]) ^ init_P[i];

653         /* Go bcopy go!  (Hope that Ultra's bcopy is faster than me!) */
654         bcopy(init_S, S, sizeof (init_S));

656         /*
657          * When initializing P and S boxes, store the results of a single
658          * encrypt-block operation in "host order", which on little-endian
659          * means byte-swapping.  Fortunately, the ntohl() function does this
660          * quite nicely, and it a NOP on big-endian machine.
661          */
662         initp = P;
663         for (i = 0; i < 9; i++) {
664                 (void) blowfish_encrypt_block(newbie, (uint8_t *)tmpblock,
665                     (uint8_t *)tmpblock);
666                 *initp++ = ntohl(tmpblock[0]);
667                 *initp++ = ntohl(tmpblock[1]);
668         }

670         initp = S;
671         for (i = 0; i < 512; i++) {
672                 (void) blowfish_encrypt_block(newbie, (uint8_t *)tmpblock,
673                     (uint8_t *)tmpblock);
674                 *initp++ = ntohl(tmpblock[0]);
675                 *initp++ = ntohl(tmpblock[1]);
676         }
688 /* EXPORT DELETE END */
677 }

679 /*
680  * Allocate key schedule for Blowfish.
681  */
682 /* ARGSUSED */
683 void *
684 blowfish_alloc_keysched(size_t *size, int kmflag)
685 {
698 /* EXPORT DELETE START */
686         keysched_t *keysched;

688 #ifdef _KERNEL
689         keysched = (keysched_t *)kmem_alloc(sizeof (keysched_t), kmflag);
690 #else
691         keysched = (keysched_t *)malloc(sizeof (keysched_t));
692 #endif /* _KERNEL */
693         if (keysched != NULL) {
694                 *size = sizeof (keysched_t);
695                 return (keysched);
696         }
710 /* EXPORT DELETE END */

698         return (NULL);
699 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1041 Thu Jul 11 01:29:06 2013
new/usr/src/common/crypto/des/Makefile
first pass
**********************************************************
```

```
#
# 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 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident  "%Z%%M% %I%     %E% SMI"
#
# common/crypto/des/Makefile
#
# include global definitions
include $(SRC)/Makefile.master

.KEEP_STATE:

FRC:

# EXPORT DELETE START
EXPORT_SRC:
        $(RM) Makefile+ des_impl.c+ des_impl.h+ des_ks.c+ sun4u/des_crypt_asm.s+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < des_impl.c > des_impl.c+
        $(MV) des_impl.c+ des_impl.c
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < des_ks.c > des_ks.c+
        $(MV) des_ks.c+ des_ks.c
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < sun4u/des_crypt_asm.s > sun4u/des_crypt_asm.s+
        $(MV) sun4u/des_crypt_asm.s+ sun4u/des_crypt_asm.s
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(RM) Makefile
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile des_impl.c des_impl.h des_ks.c sun4u/des_crypt_asm

# EXPORT DELETE END
```

```
**********************************************************
   47098 Thu Jul 11 01:29:06 2013
new/usr/src/common/crypto/des/des_impl.c
first pass
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  23  * Use is subject to license terms.
  24  */

  26 #include <sys/types.h>
  27 #include <sys/systm.h>
  28 #include <sys/ddi.h>
  29 #include <sys/sysmacros.h>
  30 #include <sys/strsun.h>
  31 #include <sys/crypto/spi.h>
  32 #include <modes/modes.h>
  33 #include <sys/crypto/common.h>
  34 #include "des_impl.h"
  35 #ifndef _KERNEL
  36 #include <strings.h>
  37 #include <stdlib.h>
  38 #endif  /* !_KERNEL */

  40 #if defined(__i386) || defined(__amd64)
  41 #include <sys/byteorder.h>
  42 #define UNALIGNED_POINTERS_PERMITTED
  43 #endif

  45 /* EXPORT DELETE START */

  45 typedef struct keysched_s {
  46         uint64_t ksch_encrypt[16];
  47         uint64_t ksch_decrypt[16];
  48 } keysched_t;
_____unchanged_portion_omitted_
 500 #endif /* !sun4u */

 504 /* EXPORT DELETE END */

 502 int
 503 des3_crunch_block(const void *cookie, const uint8_t block[DES_BLOCK_LEN],
 504     uint8_t out_block[DES_BLOCK_LEN], boolean_t decrypt)
 505 {
 510 /* EXPORT DELETE START */
 506         keysched3_t *ksch = (keysched3_t *)cookie;
```



```
 508         /*
 509          * The code below, that is always executed on LITTLE_ENDIAN machines,
 510          * reverses bytes in the block.  On BIG_ENDIAN, the same code
 511          * copies the block without reversing bytes.
 512          */
 513 #ifdef _BIG_ENDIAN
 514         if (IS_P2ALIGNED(block, sizeof (uint64_t)) &&
 515             IS_P2ALIGNED(out_block, sizeof (uint64_t))) {
 516                 if (decrypt == B_TRUE)
 517                         /* LINTED */
 518                         *(uint64_t *)out_block = des_crypt_impl(
 519                             ksch->ksch_decrypt, /* LINTED */
 520                             *(uint64_t *)block, 3);
 521                 else
 522                         /* LINTED */
 523                         *(uint64_t *)out_block = des_crypt_impl(
 524                             ksch->ksch_encrypt, /* LINTED */
 525                             *(uint64_t *)block, 3);
 526         } else
 527 #endif  /* _BIG_ENDIAN */
 528         {
 529                 uint64_t tmp;

 531 #ifdef UNALIGNED_POINTERS_PERMITTED
 532                 tmp = htonll(*(uint64_t *)(void *)&block[0]);
 533 #else
 534                 tmp = (((uint64_t)block[0] << 56) | ((uint64_t)block[1] << 48) |
 535                     ((uint64_t)block[2] << 40) | ((uint64_t)block[3] << 32) |
 536                     ((uint64_t)block[4] << 24) | ((uint64_t)block[5] << 16) |
 537                     ((uint64_t)block[6] << 8) | (uint64_t)block[7]);
 538 #endif  /* UNALIGNED_POINTERS_PERMITTED */

 540                 if (decrypt == B_TRUE)
 541                         tmp = des_crypt_impl(ksch->ksch_decrypt, tmp, 3);
 542                 else
 543                         tmp = des_crypt_impl(ksch->ksch_encrypt, tmp, 3);

 545 #ifdef UNALIGNED_POINTERS_PERMITTED
 546                 *(uint64_t *)(void *)&out_block[0] = htonll(tmp);
 547 #else
 548                 out_block[0] = tmp >> 56;
 549                 out_block[1] = tmp >> 48;
 550                 out_block[2] = tmp >> 40;
 551                 out_block[3] = tmp >> 32;
 552                 out_block[4] = tmp >> 24;
 553                 out_block[5] = tmp >> 16;
 554                 out_block[6] = tmp >> 8;
 555                 out_block[7] = (uint8_t)tmp;
 556 #endif  /* UNALIGNED_POINTERS_PERMITTED */
 557         }
 563 /* EXPORT DELETE END */
 558         return (CRYPTO_SUCCESS);
 559 }

 561 int
 562 des_crunch_block(const void *cookie, const uint8_t block[DES_BLOCK_LEN],
 563     uint8_t out_block[DES_BLOCK_LEN], boolean_t decrypt)
 564 {
 571 /* EXPORT DELETE START */
 565         keysched_t *ksch = (keysched_t *)cookie;

 567         /*
 568          * The code below, that is always executed on LITTLE_ENDIAN machines,
 569          * reverses bytes in the block.  On BIG_ENDIAN, the same code
 570          * copies the block without reversing bytes.
```

```
571          */
572 #ifdef _BIG_ENDIAN
573         if (IS_P2ALIGNED(block, sizeof (uint64_t)) &&
574             IS_P2ALIGNED(out_block, sizeof (uint64_t))) {
575                 if (decrypt == B_TRUE)
576                         /* LINTED */
577                         *(uint64_t *)out_block = des_crypt_impl(
578                             ksch->ksch_decrypt, /* LINTED */
579                             *(uint64_t *)block, 1);
580                 else
581                         /* LINTED */
582                         *(uint64_t *)out_block = des_crypt_impl(
583                             ksch->ksch_encrypt, /* LINTED */
584                             *(uint64_t *)block, 1);

586         } else
587 #endif  /* _BIG_ENDIAN */
588         {
589                 uint64_t tmp;

591 #ifdef UNALIGNED_POINTERS_PERMITTED
592                 tmp = htonll(*(uint64_t *)(void *)&block[0]);
593 #else
594                 tmp = (((uint64_t)block[0] << 56) | ((uint64_t)block[1] << 48) |
595                     ((uint64_t)block[2] << 40) | ((uint64_t)block[3] << 32) |
596                     ((uint64_t)block[4] << 24) | ((uint64_t)block[5] << 16) |
597                     ((uint64_t)block[6] << 8) | (uint64_t)block[7]);
598 #endif  /* UNALIGNED_POINTERS_PERMITTED */


601                 if (decrypt == B_TRUE)
602                         tmp = des_crypt_impl(ksch->ksch_decrypt, tmp, 1);
603                 else
604                         tmp = des_crypt_impl(ksch->ksch_encrypt, tmp, 1);

606 #ifdef UNALIGNED_POINTERS_PERMITTED
607                 *(uint64_t *)(void *)&out_block[0] = htonll(tmp);
608 #else
609                 out_block[0] = tmp >> 56;
610                 out_block[1] = tmp >> 48;
611                 out_block[2] = tmp >> 40;
612                 out_block[3] = tmp >> 32;
613                 out_block[4] = tmp >> 24;
614                 out_block[5] = tmp >> 16;
615                 out_block[6] = tmp >> 8;
616                 out_block[7] = (uint8_t)tmp;
617 #endif  /* UNALIGNED_POINTERS_PERMITTED */
618         }
626 /* EXPORT DELETE END */
619         return (CRYPTO_SUCCESS);
620 }

622 static boolean_t
623 keycheck(uint8_t *key, uint8_t *corrected_key)
624 {
633 /* EXPORT DELETE START */
625         uint64_t key_so_far;
626         uint_t i;
627         /*
628          * Table of weak and semi-weak keys.  Fortunately, weak keys are
629          * endian-independent, and some semi-weak keys can be paired up in
630          * endian-opposite order.  Since keys are stored as uint64_t's,
631          * use the ifdef _LITTLE_ENDIAN where appropriate.
632          */
633         static uint64_t des_weak_keys[] = {
634                 /* Really weak keys.  Byte-order independent values. */
```



```
635                 0x0101010101010101ULL,
636                 0x1f1f1f1f0e0e0e0eULL,
637                 0xe0e0e0e0f1f1f1f1ULL,
638                 0xfefefefefefefefeULL,

640                 /* Semi-weak (and a few possibly-weak) keys. */

642                 /* Byte-order independent semi-weak keys. */
643                 0x01fe01fe01fe01feULL,  0xfe01fe01fe01fe01ULL,

645                 /* Byte-order dependent semi-weak keys. */
646 #ifdef _LITTLE_ENDIAN
647                 0xf10ef10ee01fe01fULL,  0x0ef10ef11fe01fe0ULL,
648                 0x01f101f101e001e0ULL,  0xf101f101e001e001ULL,
649                 0x0efe0efe1ffe1ffeULL,  0xfe0efe0efe1ffe1fULL,
650                 0x010e010e011f011fULL,  0x0e010e011f011f01ULL,
651                 0xf1fef1fee0fee0feULL,  0xfef1fef1fee0fee0ULL,
652 #else   /* Big endian */
653                 0x1fe01fe00ef10ef1ULL,  0xe01fe01ff10ef10eULL,
654                 0x01e001e001f101f1ULL,  0xe001e001f101f101ULL,
655                 0x1ffe1ffe0efe0efeULL,  0xfe1ffe1ffe0efe0eULL,
656                 0x011f011f010e010eULL,  0x1f011f010e010e01ULL,
657                 0xe0fee0fef1fef1feULL,  0xfee0fee0fef1fef1ULL,
658 #endif  /* _LITTLE_ENDIAN */

660                 /* We'll save the other possibly-weak keys for the future. */
661         };

663         if (key == NULL)
664                 return (B_FALSE);

666 #ifdef UNALIGNED_POINTERS_PERMITTED
667         key_so_far = htonll(*(uint64_t *)(void *)&key[0]);
668 #else
669         /*
670          * The code below reverses the bytes on LITTLE_ENDIAN machines.
671          * On BIG_ENDIAN, the same code copies without reversing
672          * the bytes.
673          */
674         key_so_far = (((uint64_t)key[0] << 56) | ((uint64_t)key[1] << 48) |
675             ((uint64_t)key[2] << 40) | ((uint64_t)key[3] << 32) |
676             ((uint64_t)key[4] << 24) | ((uint64_t)key[5] << 16) |
677             ((uint64_t)key[6] << 8) | (uint64_t)key[7]);
678 #endif  /* UNALIGNED_POINTERS_PERMITTED */

680         /*
681          * Fix parity.
682          */
683         fix_des_parity(&key_so_far);

685         /* Do weak key check itself. */
686         for (i = 0; i < (sizeof (des_weak_keys) / sizeof (uint64_t)); i++)
687                 if (key_so_far == des_weak_keys[i]) {
688                         return (B_FALSE);
689                 }

691         if (corrected_key != NULL) {
692 #ifdef UNALIGNED_POINTERS_PERMITTED
693                 *(uint64_t *)(void *)&corrected_key[0] = htonll(key_so_far);
694 #else
695                 /*
696                  * The code below reverses the bytes on LITTLE_ENDIAN machines.
697                  * On BIG_ENDIAN, the same code copies without reversing
698                  * the bytes.
699                  */
700                 corrected_key[0] = key_so_far >> 56;
```

```
701                 corrected_key[1] = key_so_far >> 48;
702                 corrected_key[2] = key_so_far >> 40;
703                 corrected_key[3] = key_so_far >> 32;
704                 corrected_key[4] = key_so_far >> 24;
705                 corrected_key[5] = key_so_far >> 16;
706                 corrected_key[6] = key_so_far >> 8;
707                 corrected_key[7] = (uint8_t)key_so_far;
708 #endif  /* UNALIGNED_POINTERS_PERMITTED */
709         }
719 /* EXPORT DELETE END */
710         return (B_TRUE);
711 }

713 static boolean_t
714 des23_keycheck(uint8_t *key, uint8_t *corrected_key, boolean_t des3)
715 {
726 /* EXPORT DELETE START */
716         uint64_t aligned_key[DES3_KEYSIZE / sizeof (uint64_t)];
717         uint64_t key_so_far, scratch, *currentkey;
718         uint_t j, num_weakkeys = 0;
719         uint8_t keysize = DES3_KEYSIZE;
720         uint8_t checks = 3;

722         if (key == NULL) {
723                 return (B_FALSE);
724         }

726         if (des3 == B_FALSE) {
727                 keysize = DES2_KEYSIZE;
728                 checks = 2;
729         }

731         if (!IS_P2ALIGNED(key, sizeof (uint64_t))) {
732                 bcopy(key, aligned_key, keysize);
733                 currentkey = (uint64_t *)aligned_key;
734         } else {
735                 /* LINTED */
736                 currentkey = (uint64_t *)key;
737         }

739         for (j = 0; j < checks; j++) {
740                 key_so_far = currentkey[j];

742                 if (!keycheck((uint8_t *)&key_so_far, (uint8_t *)&scratch)) {
743                         if (++num_weakkeys > 1) {
744                                 return (B_FALSE);
745                         }
746                         /*
747                          * We found a weak key, but since
748                          * we've only found one weak key,
749                          * we can not reject the whole 3DES
750                          * set of keys as weak.
751                          *
752                          * Break from the weak key loop
753                          * (since this DES key is weak) and
754                          * continue on.
755                          */
756                 }

758                 currentkey[j] = scratch;
759         }

761         /*
762          * Perform key equivalence checks, now that parity is properly set.
763          * 1st and 2nd keys must be unique, the 3rd key can be the same as
764          * the 1st key for the 2 key variant of 3DES.
```



```
765          */
766         if (currentkey[0] == currentkey[1] || currentkey[1] == currentkey[2])
767                 return (B_FALSE);

769         if (corrected_key != NULL) {
770                 bcopy(currentkey, corrected_key, keysize);
771         }

784 /* EXPORT DELETE END */
773         return (B_TRUE);
774 }
```

_____***unchanged_portion_omitted_***

```
790 void
791 des_parity_fix(uint8_t *key, des_strength_t strength, uint8_t *corrected_key)
792 {
805 /* EXPORT DELETE START */
793         uint64_t aligned_key[DES3_KEYSIZE / sizeof (uint64_t)];
794         uint8_t *paritied_key;
795         uint64_t key_so_far;
796         int i = 0, offset = 0;

798         if (strength == DES)
799                 bcopy(key, aligned_key, DES_KEYSIZE);
800         else
801                 bcopy(key, aligned_key, DES3_KEYSIZE);

803         paritied_key = (uint8_t *)aligned_key;
804         while (strength > i) {
805                 offset = 8 * i;
806 #ifdef UNALIGNED_POINTERS_PERMITTED
807                 key_so_far = htonll(*(uint64_t *)(void *)&paritied_key[offset]);
808 #else
809                 key_so_far = (((uint64_t)paritied_key[offset + 0] << 56) |
810                     ((uint64_t)paritied_key[offset + 1] << 48) |
811                     ((uint64_t)paritied_key[offset + 2] << 40) |
812                     ((uint64_t)paritied_key[offset + 3] << 32) |
813                     ((uint64_t)paritied_key[offset + 4] << 24) |
814                     ((uint64_t)paritied_key[offset + 5] << 16) |
815                     ((uint64_t)paritied_key[offset + 6] << 8) |
816                     (uint64_t)paritied_key[offset + 7]);
817 #endif  /* UNALIGNED_POINTERS_PERMITTED */

819                 fix_des_parity(&key_so_far);

821 #ifdef UNALIGNED_POINTERS_PERMITTED
822                 *(uint64_t *)(void *)&paritied_key[offset] = htonll(key_so_far);
823 #else
824                 paritied_key[offset + 0] = key_so_far >> 56;
825                 paritied_key[offset + 1] = key_so_far >> 48;
826                 paritied_key[offset + 2] = key_so_far >> 40;
827                 paritied_key[offset + 3] = key_so_far >> 32;
828                 paritied_key[offset + 4] = key_so_far >> 24;
829                 paritied_key[offset + 5] = key_so_far >> 16;
830                 paritied_key[offset + 6] = key_so_far >> 8;
831                 paritied_key[offset + 7] = (uint8_t)key_so_far;
832 #endif  /* UNALIGNED_POINTERS_PERMITTED */

834                 i++;
835         }

837         bcopy(paritied_key, corrected_key, DES_KEYSIZE * strength);
851 /* EXPORT DELETE END */
838 }
```

```
841 /*
842  * Initialize key schedule for DES, DES2, and DES3
843  */
844 void
845 des_init_keysched(uint8_t *cipherKey, des_strength_t strength, void *ks)
846 {
861 /* EXPORT DELETE START */
847         uint64_t *encryption_ks;
848         uint64_t *decryption_ks;
849         uint64_t keysched[48];
850         uint64_t key_uint64[3];
851         uint64_t tmp;
852         uint_t keysize, i, j;

854         switch (strength) {
855         case DES:
856                 keysize = DES_KEYSIZE;
857                 encryption_ks = ((keysched_t *)ks)->ksch_encrypt;
858                 decryption_ks = ((keysched_t *)ks)->ksch_decrypt;
859                 break;
860         case DES2:
861                 keysize = DES2_KEYSIZE;
862                 encryption_ks = ((keysched3_t *)ks)->ksch_encrypt;
863                 decryption_ks = ((keysched3_t *)ks)->ksch_decrypt;
864                 break;
865         case DES3:
866                 keysize = DES3_KEYSIZE;
867                 encryption_ks = ((keysched3_t *)ks)->ksch_encrypt;
868                 decryption_ks = ((keysched3_t *)ks)->ksch_decrypt;
869         }

871         /*
872          * The code below, that is always executed on LITTLE_ENDIAN machines,
873          * reverses every 8 bytes in the key.  On BIG_ENDIAN, the same code
874          * copies the key without reversing bytes.
875          */
876 #ifdef _BIG_ENDIAN
877         if (IS_P2ALIGNED(cipherKey, sizeof (uint64_t))) {
878                 for (i = 0, j = 0; j < keysize; i++, j += 8) {
879                         /* LINTED: pointer alignment */
880                         key_uint64[i] = *((uint64_t *)&cipherKey[j]);
881                 }
882         } else
883 #endif  /* _BIG_ENDIAN */
884         {
885                 for (i = 0, j = 0; j < keysize; i++, j += 8) {
886 #ifdef UNALIGNED_POINTERS_PERMITTED
887                         key_uint64[i] =
888                             htonll(*(uint64_t *)(void *)&cipherKey[j]);
889 #else
890                         key_uint64[i] = (((uint64_t)cipherKey[j] << 56) |
891                             ((uint64_t)cipherKey[j + 1] << 48) |
892                             ((uint64_t)cipherKey[j + 2] << 40) |
893                             ((uint64_t)cipherKey[j + 3] << 32) |
894                             ((uint64_t)cipherKey[j + 4] << 24) |
895                             ((uint64_t)cipherKey[j + 5] << 16) |
896                             ((uint64_t)cipherKey[j + 6] << 8) |
897                             (uint64_t)cipherKey[j + 7]);
898 #endif  /* UNALIGNED_POINTERS_PERMITTED */
899                 }
900         }

902         switch (strength) {
903         case DES:
904                 des_ks(keysched, key_uint64[0]);
905                 break;
```



```
907         case DES2:
908                 /* DES2 is just DES3 with the first and third keys the same */
909                 bcopy(key_uint64, key_uint64 + 2, DES_KEYSIZE);
910                 /* FALLTHRU */
911         case DES3:
912                 des_ks(keysched, key_uint64[0]);
913                 des_ks(keysched + 16, key_uint64[1]);
914                 for (i = 0; i < 8; i++) {
915                         tmp = keysched[16+i];
916                         keysched[16+i] = keysched[31-i];
917                         keysched[31-i] = tmp;
918                 }
919                 des_ks(keysched+32, key_uint64[2]);
920                 keysize = DES3_KEYSIZE;
921         }

923         /* save the encryption keyschedule */
924         bcopy(keysched, encryption_ks, keysize * 16);

926         /* reverse the key schedule */
927         for (i = 0; i < keysize; i++) {
928                 tmp = keysched[i];
929                 keysched[i] = keysched[2 * keysize - 1 - i];
930                 keysched[2 * keysize -1 -i] = tmp;
931         }

933         /* save the decryption keyschedule */
934         bcopy(keysched, decryption_ks, keysize * 16);
950 /* EXPORT DELETE END */
935 }

937 /*
938  * Allocate key schedule.
939  */
940 /*ARGSUSED*/
941 void *
942 des_alloc_keysched(size_t *keysched_size, des_strength_t strength, int kmflag)
943 {
944         void *keysched;

962 /* EXPORT DELETE START */

946         size_t size;

948         switch (strength) {
949         case DES:
950                 size = sizeof (keysched_t);
951                 break;
952         case DES2:
953         case DES3:
954                 size = sizeof (keysched3_t);
955         }

957 #ifdef  _KERNEL
958         keysched = (keysched_t *)kmem_alloc(size, kmflag);
959 #else   /* !_KERNEL */
960         keysched = (keysched_t *)malloc(size);
961 #endif  /* _KERNEL */

963         if (keysched == NULL)
964                 return (NULL);

966         if (keysched_size != NULL)
967                 *keysched_size = size;
```

```
 987 /* EXPORT DELETE END */

 969         return (keysched);
 970 }

 972 /*
 973  * Replace the LSB of each byte by the xor of the other
 974  * 7 bits.  The tricky thing is that the original contents of the LSBs
 975  * are nullified by including them twice in the xor computation.
 976  */
 977 static void
 978 fix_des_parity(uint64_t *keyp)
 979 {
1000 /* EXPORT DELETE START */
 980         uint64_t k = *keyp;
 981         k ^= k >> 1;
 982         k ^= k >> 2;
 983         k ^= k >> 4;
 984         *keyp ^= (k & 0x0101010101010101ULL);
 985         *keyp ^= 0x0101010101010101ULL;
1007 /* EXPORT DELETE END */
 986 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   13555 Thu Jul 11 01:29:07 2013
new/usr/src/common/crypto/des/des_ks.c
first pass
**********************************************************
```

```
/*
 * 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.
 */

#include <sys/types.h>

#pragma ident   "%Z%%M% %I%     %E% SMI"

/* EXPORT DELETE START */

static uint64_t pc1_table[2][128]=
{
/* BEGIN CSTYLED */
{
0x0000000000000000ULL, 0x0000000000100000ULL, 0x0000000000001000ULL, 0x000000000
0x0000000000000010ULL, 0x0000000000100010ULL, 0x0000000000001010ULL, 0x000000000
0x0000000000000001ULL, 0x0000000000100001ULL, 0x0000000000001001ULL, 0x000000000
0x0000000000000011ULL, 0x0000000000100011ULL, 0x0000000000001011ULL, 0x000000000
0x0000000100000000ULL, 0x0000000100100000ULL, 0x0000000100001000ULL, 0x000000010
0x0000000100000010ULL, 0x0000000100100010ULL, 0x0000000100001010ULL, 0x000000010
0x0000000100000001ULL, 0x0000000100100001ULL, 0x0000000100001001ULL, 0x000000010
0x0000000100000011ULL, 0x0000000100100011ULL, 0x0000000100001011ULL, 0x000000010
0x0000010000000000ULL, 0x0000010000100000ULL, 0x0000010000001000ULL, 0x000001000
0x0000010000000010ULL, 0x0000010000100010ULL, 0x0000010000001010ULL, 0x000001000
0x0000010000000001ULL, 0x0000010000100001ULL, 0x0000010000001001ULL, 0x000001000
0x0000010000000011ULL, 0x0000010000100011ULL, 0x0000010000001011ULL, 0x000001000
0x0000010100000000ULL, 0x0000010100100000ULL, 0x0000010100001000ULL, 0x000001010
0x0000010100000010ULL, 0x0000010100100010ULL, 0x0000010100001010ULL, 0x000001010
0x0000010100000001ULL, 0x0000010100100001ULL, 0x0000010100001001ULL, 0x000001010
0x0000010100000011ULL, 0x0000010100100011ULL, 0x0000010100001011ULL, 0x000001010
0x0001000000000000ULL, 0x0001000000100000ULL, 0x0001000000001000ULL, 0x000100000
0x0001000000000010ULL, 0x0001000000100010ULL, 0x0001000000001010ULL, 0x000100000
0x0001000000000001ULL, 0x0001000000100001ULL, 0x0001000000001001ULL, 0x000100000
0x0001000000000011ULL, 0x0001000000100011ULL, 0x0001000000001011ULL, 0x000100000
0x0001000100000000ULL, 0x0001000100100000ULL, 0x0001000100001000ULL, 0x000100010
0x0001000100000010ULL, 0x0001000100100010ULL, 0x0001000100001010ULL, 0x000100010
0x0001000100000001ULL, 0x0001000100100001ULL, 0x0001000100001001ULL, 0x000100010
0x0001000100000011ULL, 0x0001000100100011ULL, 0x0001000100001011ULL, 0x000100010
0x0001010000000000ULL, 0x0001010000100000ULL, 0x0001010000001000ULL, 0x000101000
```



```
0x0001010000000010ULL, 0x0001010000100010ULL, 0x0001010000001010ULL, 0x000101000
0x0001010000000001ULL, 0x0001010000100001ULL, 0x0001010000001001ULL, 0x000101000
0x0001010000000011ULL, 0x0001010000100011ULL, 0x0001010000001011ULL, 0x000101000
0x0001010100000000ULL, 0x0001010100100000ULL, 0x0001010100001000ULL, 0x000101010
0x0001010100000010ULL, 0x0001010100100010ULL, 0x0001010100001010ULL, 0x000101010
0x0001010100000001ULL, 0x0001010100100001ULL, 0x0001010100001001ULL, 0x000101010
0x0001010100000011ULL, 0x0001010100100011ULL, 0x0001010100001011ULL, 0x000101010
},
_____unchanged_portion_omitted_

/* EXPORT DELETE END */
```

```
**********************************************************
   56642 Thu Jul 11 01:29:08 2013
new/usr/src/common/crypto/des/sun4u/des_crypt_asm.s
first pass
**********************************************************
```

```
/*
 * 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 2004 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma ident   "%Z%%M% %I%     %E% SMI"

/*
 * Unified version for both position independent and non position independent
 * for both v8plus and v9
 * compile with:
 *
 * cc -c -xarch=v8plus des_crypt_asm.s     or
 * cc -c -arch=v9 des_crypt_asm.s
 * for kernel use (no -KPIC)
 *
 * and with
 *
 * cc -c -xarch=v8plus -KPIC -DPIC des_crypt_asm.s     or
 * cc -c -arch=v9 -KPIC -DPIC des_crypt_asm.s
 * for  .so  use
 *
 * EXPORT DELETE START
 *
 * The tables were generated by a C program, compiled into the C version
 * of this function, from which a .s was generated by the C compiler and
 * that .s was used as a starting point for this one, in particular for
 * the data definitions. It is important, though that the tables and
 * the code both remain in the text section and in this order, otherwise,
 * at least on UltraSparc-II processors, collisions in the E-cache are
 * highly probable between the code and the data it is using which can
 * result in up to 40% performance loss
 *
 * For a description of the DES algithm, see NIST publication FIPS PUB 46-3
 *
 * In this implementation, the 16 rounds of DES are carried out by unrolling
 * a loop that computes two rounds. For those 2 rounds, the two parts of
 * the intermediate variable (L and R in the FIPS pub) are kept in their
 * extended forms (i.e. in the one after applying the transformation E),
 * with the appropriate bits repeated so that bits needed for the S-box
```



```
 * lookups are in consecutive positions. So the bits of the L (or R)
 * variable appear in the following order (X represents a bit that is not
 * from L (R), these bits are always 0):
 * 32  1  2  3  4  5  X  X   X  X  X  X  X  4  5
 *  6  7  8  9  8  9 10 11  12 13 12 13 14 15 16 17
 * 16 17 18 19 20 21  X  X   X  X  X 20 21 22 23 24
 * 25 24 25 26 27 28 29 28  29 30 31 32  1  X  X  X
 * This arrangement makes it possible that 3 of the 8 S-box indices
 * can be extracted by a single instruction: srlx by 55 for the S1 index,
 * srl by 23 for the S5 index and and by 0x1f80 for the S8 index. The rest
 * of the indices requires two operations, a shift and an and.
 * The tables for the S-boxes are computed in such a way that when or-ed
 * together, they give the result of the S-box, P and E computations.
 * Also, the key schedule bits are computed to follow this bit-scheme.
 * The initial permutation tables are also computed to produce this
 * bit distribution and the final permutation works from these, too.
 *
 * The end of each round is overlapped with the beginning of the next
 * one since after the first 6 S-box lookups all the bits necessary
 * for one S-box lookup in the next round can be computed (by xor-ing
 * the next key schedule item to the partially computed next R).
 *
 * EXPORT DELETE END
 */

#if defined(lint) || defined(__lint)
        /* LINTED */
        /* Nothing to be linted in this file, its pure assembly source */
#else   /* lint || __lint */

        .register       %g2,#scratch
        .register       %g3,#scratch

        .file   "encrypt_asm.S"

        .section        ".text",#alloc
        .align  32

/* EXPORT DELETE START */

!
! CONSTANT POOL
!

des_sbox_table:
        .word   5121
        .word   1073872896
        .word   0
        .word   0
        .word   1
        .word   1073741824
        .word   5121
        .word   1073872928
        .word   5121
        .word   1073741856
        .word   1
        .word   1073872928
        .word   0
        .word   32
        .word   1
        .word   1073741824
        .word   0
        .word   131072
        .word   5121
        .word   1073872896
        .word   5121
```

```
122         .word   1073872928
123         .word   0
124         .word   131072
125         .word   5120
126         .word   131104
127         .word   5121
128         .word   1073741856
129         .word   5120
130         .word   0
131         .word   0
132         .word   32
133         .word   0
134         .word   131104
135         .word   5120
136         .word   131072
137         .word   5120
138         .word   131072
139         .word   1
140         .word   1073872896
141         .word   1
142         .word   1073872896
143         .word   5121
144         .word   1073741824
145         .word   5121
146         .word   1073741824
147         .word   5120
148         .word   131104
149         .word   1
150         .word   1073741856
151         .word   5120
152         .word   32
153         .word   5120
154         .word   32
155         .word   1
156         .word   1073741856
157         .word   0
158         .word   0
159         .word   0
160         .word   131104
161         .word   1
162         .word   1073872928
163         .word   5120
164         .word   0
165         .word   1
166         .word   1073741824
167         .word   5121
168         .word   1073872928
169         .word   0
170         .word   32
171         .word   5121
172         .word   1073741824
173         .word   5121
174         .word   1073872896
175         .word   5120
176         .word   0
177         .word   5120
178         .word   0
179         .word   0
180         .word   131072
181         .word   5121
182         .word   1073741856
183         .word   1
184         .word   1073741824
185         .word   1
186         .word   1073872896
187         .word   5120
```



```
188         .word   32
189         .word   0
190         .word   131072
191         .word   0
192         .word   32
193         .word   5120
194         .word   131104
195         .word   1
196         .word   1073872928
197         .word   5121
198         .word   1073872928
199         .word   1
200         .word   1073741856
201         .word   5121
202         .word   1073741824
203         .word   5120
204         .word   131104
205         .word   5120
206         .word   32
207         .word   0
208         .word   131104
209         .word   1
210         .word   1073872928
211         .word   5121
212         .word   1073872896
213         .word   0
214         .word   131104
215         .word   5120
216         .word   131072
217         .word   5120
218         .word   131072
219         .word   0
220         .word   0
221         .word   1
222         .word   1073741856
223         .word   1
224         .word   1073872896
225         .word   0
226         .word   0
227         .word   5121
228         .word   1073741856
229         .word   536870992
230         .word   536872192
231         .word   536870912
232         .word   536870912
233         .word   0
234         .word   536870912
235         .word   80
236         .word   536872192
237         .word   80
238         .word   0
239         .word   0
240         .word   1280
241         .word   536870992
242         .word   1280
243         .word   536870912
244         .word   536872192
245         .word   536870912
246         .word   1280
247         .word   536870992
248         .word   536872192
249         .word   536870992
250         .word   536870912
251         .word   536870912
252         .word   0
253         .word   536870912
```

```
254         .word   536870912
255         .word   80
256         .word   0
257         .word   0
258         .word   1280
259         .word   536870992
260         .word   1280
261         .word   80
262         .word   536870912
263         .word   80
264         .word   1280
265         .word   536870912
266         .word   536872192
267         .word   0
268         .word   0
269         .word   536870912
270         .word   0
271         .word   0
272         .word   536870912
273         .word   80
274         .word   536872192
275         .word   536870992
276         .word   0
277         .word   80
278         .word   1280
279         .word   536870912
280         .word   1280
281         .word   0
282         .word   0
283         .word   80
284         .word   536870912
285         .word   0
286         .word   536872192
287         .word   536870992
288         .word   536870912
289         .word   536870992
290         .word   0
291         .word   0
292         .word   536872192
293         .word   0
294         .word   0
295         .word   80
296         .word   536872192
297         .word   536870992
298         .word   1280
299         .word   80
300         .word   0
301         .word   536870912
302         .word   536872192
303         .word   536870992
304         .word   0
305         .word   536870992
306         .word   536870912
307         .word   0
308         .word   536870912
309         .word   536870992
310         .word   0
311         .word   536870912
312         .word   536870912
313         .word   0
314         .word   1280
315         .word   536870992
316         .word   536872192
317         .word   80
318         .word   536872192
319         .word   0
```



```
320         .word   1280
321         .word   0
322         .word   536870912
323         .word   536870912
324         .word   0
325         .word   0
326         .word   536872192
327         .word   536870992
328         .word   536870912
329         .word   80
330         .word   0
331         .word   536870912
332         .word   1280
333         .word   80
334         .word   1280
335         .word   536870912
336         .word   536872192
337         .word   536870912
338         .word   1280
339         .word   80
340         .word   1280
341         .word   80
342         .word   536870912
343         .word   0
344         .word   0
345         .word   536870912
346         .word   536870912
347         .word   0
348         .word   536872192
349         .word   536870912
350         .word   0
351         .word   536870992
352         .word   1280
353         .word   536870992
354         .word   536872192
355         .word   80
356         .word   536870912
357         .word   0
358         .word   81984
359         .word   32770
360         .word   -2147401728
361         .word   0
362         .word   0
363         .word   32770
364         .word   -2147483584
365         .word   32768
366         .word   81920
367         .word   0
368         .word   0
369         .word   2
370         .word   -2147401664
371         .word   32768
372         .word   81920
373         .word   2
374         .word   -2147483584
375         .word   32768
376         .word   64
377         .word   32768
378         .word   64
379         .word   2
380         .word   -2147483648
381         .word   32770
382         .word   -2147401664
383         .word   2
384         .word   -2147483584
385         .word   32770
```

```
386         .word   -2147483648
387         .word   0
388         .word   81984
389         .word   32768
390         .word   0
391         .word   0
392         .word   64
393         .word   32770
394         .word   -2147401728
395         .word   0
396         .word   81920
397         .word   2
398         .word   -2147401728
399         .word   32770
400         .word   -2147483648
401         .word   32770
402         .word   -2147483584
403         .word   2
404         .word   -2147401664
405         .word   32768
406         .word   81984
407         .word   2
408         .word   -2147401728
409         .word   2
410         .word   -2147483648
411         .word   32768
412         .word   81984
413         .word   0
414         .word   64
415         .word   32770
416         .word   -2147401664
417         .word   0
418         .word   81920
419         .word   32768
420         .word   0
421         .word   32770
422         .word   -2147401728
423         .word   32768
424         .word   0
425         .word   2
426         .word   -2147483584
427         .word   0
428         .word   81984
429         .word   2
430         .word   -2147483648
431         .word   32770
432         .word   -2147401728
433         .word   32768
434         .word   81920
435         .word   0
436         .word   0
437         .word   0
438         .word   81920
439         .word   2
440         .word   -2147483584
441         .word   32770
442         .word   -2147401664
443         .word   32768
444         .word   81920
445         .word   32768
446         .word   64
447         .word   0
448         .word   81920
449         .word   0
450         .word   0
451         .word   32770
```



```
452         .word   -2147483584
453         .word   32768
454         .word   81984
455         .word   2
456         .word   -2147483648
457         .word   32768
458         .word   0
459         .word   32770
460         .word   -2147401664
461         .word   0
462         .word   64
463         .word   2
464         .word   -2147401664
465         .word   2
466         .word   -2147401728
467         .word   32768
468         .word   64
469         .word   32770
470         .word   -2147483648
471         .word   32768
472         .word   81984
473         .word   0
474         .word   81984
475         .word   32770
476         .word   -2147483648
477         .word   2
478         .word   -2147401664
479         .word   0
480         .word   64
481         .word   32770
482         .word   -2147483584
483         .word   2
484         .word   -2147401728
485         .word   1073742336
486         .word   135266312
487         .word   1073741824
488         .word   135270408
489         .word   1073741824
490         .word   135270408
491         .word   0
492         .word   4096
493         .word   512
494         .word   135270400
495         .word   1073742336
496         .word   4104
497         .word   1073742336
498         .word   8
499         .word   1073741824
500         .word   135266312
501         .word   0
502         .word   0
503         .word   512
504         .word   135266304
505         .word   512
506         .word   135266304
507         .word   1073742336
508         .word   135270408
509         .word   1073741824
510         .word   4104
511         .word   0
512         .word   0
513         .word   512
514         .word   4096
515         .word   1073742336
516         .word   8
517         .word   1073741824
```

```
518         .word   8
519         .word   0
520         .word   135266304
521         .word   512
522         .word   0
523         .word   1073742336
524         .word   135266312
525         .word   0
526         .word   4096
527         .word   512
528         .word   0
529         .word   1073741824
530         .word   135266312
531         .word   0
532         .word   135270400
533         .word   1073742336
534         .word   4104
535         .word   1073741824
536         .word   8
537         .word   0
538         .word   135270400
539         .word   512
540         .word   4096
541         .word   0
542         .word   135266304
543         .word   512
544         .word   135270400
545         .word   1073742336
546         .word   135270408
547         .word   1073741824
548         .word   4104
549         .word   512
550         .word   4096
551         .word   1073742336
552         .word   8
553         .word   512
554         .word   135266304
555         .word   1073742336
556         .word   135270408
557         .word   1073741824
558         .word   4104
559         .word   0
560         .word   0
561         .word   0
562         .word   0
563         .word   512
564         .word   135266304
565         .word   0
566         .word   135270400
567         .word   512
568         .word   4096
569         .word   1073742336
570         .word   4104
571         .word   1073741824
572         .word   8
573         .word   1073742336
574         .word   135266312
575         .word   1073741824
576         .word   135270408
577         .word   1073741824
578         .word   135270408
579         .word   0
580         .word   4096
581         .word   1073742336
582         .word   135270408
583         .word   1073741824
```



```
584         .word   4104
585         .word   1073741824
586         .word   8
587         .word   0
588         .word   135266304
589         .word   1073742336
590         .word   8
591         .word   1073741824
592         .word   135266312
593         .word   512
594         .word   135270400
595         .word   1073742336
596         .word   4104
597         .word   1073741824
598         .word   135266312
599         .word   0
600         .word   135270400
601         .word   512
602         .word   0
603         .word   1073742336
604         .word   135266312
605         .word   0
606         .word   4096
607         .word   512
608         .word   0
609         .word   0
610         .word   135266304
611         .word   512
612         .word   135270400
613         .word   0
614         .word   40960
615         .word   10248
616         .word   40960
617         .word   10248
618         .word   0
619         .word   268445696
620         .word   40960
621         .word   8
622         .word   0
623         .word   0
624         .word   40960
625         .word   268435456
626         .word   0
627         .word   10248
628         .word   0
629         .word   268435464
630         .word   40960
631         .word   8
632         .word   0
633         .word   10240
634         .word   40960
635         .word   268435464
636         .word   40960
637         .word   268445696
638         .word   40960
639         .word   268445704
640         .word   0
641         .word   8
642         .word   40960
643         .word   268435456
644         .word   0
645         .word   10240
646         .word   0
647         .word   268435464
648         .word   0
649         .word   268435464
```

```
650         .word   0
651         .word   0
652         .word   0
653         .word   268435456
654         .word   40960
655         .word   268445704
656         .word   40960
657         .word   268445704
658         .word   40960
659         .word   10240
660         .word   40960
661         .word   268445704
662         .word   0
663         .word   268435456
664         .word   40960
665         .word   0
666         .word   0
667         .word   268445696
668         .word   0
669         .word   10248
670         .word   40960
671         .word   10240
672         .word   0
673         .word   268445696
674         .word   0
675         .word   8
676         .word   40960
677         .word   8
678         .word   0
679         .word   268445696
680         .word   40960
681         .word   0
682         .word   40960
683         .word   10240
684         .word   0
685         .word   268435456
686         .word   0
687         .word   10248
688         .word   0
689         .word   268445696
690         .word   40960
691         .word   268435464
692         .word   40960
693         .word   10240
694         .word   40960
695         .word   268435456
696         .word   0
697         .word   268445704
698         .word   0
699         .word   10248
700         .word   40960
701         .word   268435464
702         .word   40960
703         .word   0
704         .word   40960
705         .word   10240
706         .word   0
707         .word   268445704
708         .word   0
709         .word   268445704
710         .word   40960
711         .word   8
712         .word   40960
713         .word   268445696
714         .word   0
715         .word   268445704
```



```
716         .word   40960
717         .word   10248
718         .word   0
719         .word   0
720         .word   0
721         .word   268435464
722         .word   0
723         .word   268445696
724         .word   0
725         .word   8
726         .word   40960
727         .word   10240
728         .word   40960
729         .word   268435456
730         .word   40960
731         .word   8
732         .word   0
733         .word   0
734         .word   0
735         .word   268435464
736         .word   0
737         .word   10248
738         .word   40960
739         .word   268435456
740         .word   40960
741         .word   134348800
742         .word   640
743         .word   134349056
744         .word   0
745         .word   0
746         .word   268435456
747         .word   134349056
748         .word   268436096
749         .word   134349056
750         .word   0
751         .word   0
752         .word   640
753         .word   134349056
754         .word   268436096
755         .word   256
756         .word   0
757         .word   134348800
758         .word   268435456
759         .word   256
760         .word   268436096
761         .word   256
762         .word   0
763         .word   134348800
764         .word   640
765         .word   256
766         .word   640
767         .word   134348800
768         .word   268435456
769         .word   134348800
770         .word   0
771         .word   0
772         .word   268436096
773         .word   0
774         .word   0
775         .word   256
776         .word   640
777         .word   134348800
778         .word   268436096
779         .word   0
780         .word   268435456
781         .word   256
```

```
 782         .word   268435456
 783         .word   134348800
 784         .word   268436096
 785         .word   0
 786         .word   640
 787         .word   134349056
 788         .word   640
 789         .word   134349056
 790         .word   640
 791         .word   0
 792         .word   0
 793         .word   256
 794         .word   268436096
 795         .word   134349056
 796         .word   268435456
 797         .word   0
 798         .word   268436096
 799         .word   256
 800         .word   268435456
 801         .word   134349056
 802         .word   268435456
 803         .word   134348800
 804         .word   0
 805         .word   134348800
 806         .word   268435456
 807         .word   0
 808         .word   640
 809         .word   134349056
 810         .word   640
 811         .word   256
 812         .word   268435456
 813         .word   134349056
 814         .word   268436096
 815         .word   256
 816         .word   0
 817         .word   0
 818         .word   268436096
 819         .word   134348800
 820         .word   640
 821         .word   256
 822         .word   0
 823         .word   134348800
 824         .word   268435456
 825         .word   134348800
 826         .word   0
 827         .word   0
 828         .word   268436096
 829         .word   134348800
 830         .word   640
 831         .word   134349056
 832         .word   268436096
 833         .word   256
 834         .word   268435456
 835         .word   134349056
 836         .word   0
 837         .word   256
 838         .word   268436096
 839         .word   134349056
 840         .word   268435456
 841         .word   0
 842         .word   0
 843         .word   134349056
 844         .word   640
 845         .word   0
 846         .word   640
 847         .word   0
```



```
 848         .word   268435456
 849         .word   134349056
 850         .word   0
 851         .word   256
 852         .word   268436096
 853         .word   0
 854         .word   268435456
 855         .word   256
 856         .word   640
 857         .word   134348800
 858         .word   268436096
 859         .word   0
 860         .word   0
 861         .word   134349056
 862         .word   268435456
 863         .word   134348800
 864         .word   0
 865         .word   256
 866         .word   640
 867         .word   134348800
 868         .word   268436096
 869         .word   160
 870         .word   0
 871         .word   -2147467104
 872         .word   16
 873         .word   -2147467264
 874         .word   262160
 875         .word   0
 876         .word   0
 877         .word   0
 878         .word   262144
 879         .word   -2147467264
 880         .word   262160
 881         .word   -2147483488
 882         .word   262160
 883         .word   16544
 884         .word   262144
 885         .word   -2147467104
 886         .word   262160
 887         .word   160
 888         .word   0
 889         .word   0
 890         .word   0
 891         .word   -2147467264
 892         .word   16
 893         .word   -2147483648
 894         .word   16
 895         .word   16384
 896         .word   0
 897         .word   -2147467104
 898         .word   16
 899         .word   -2147483648
 900         .word   262160
 901         .word   16384
 902         .word   262144
 903         .word   -2147483488
 904         .word   262160
 905         .word   -2147483488
 906         .word   16
 907         .word   16384
 908         .word   262144
 909         .word   -2147467264
 910         .word   16
 911         .word   16544
 912         .word   0
 913         .word   16544
```

```
 914         .word   262144
 915         .word   -2147483488
 916         .word   16
 917         .word   16544
 918         .word   0
 919         .word   0
 920         .word   262144
 921         .word   -2147483648
 922         .word   262160
 923         .word   -2147467104
 924         .word   262160
 925         .word   160
 926         .word   262144
 927         .word   -2147483648
 928         .word   16
 929         .word   16384
 930         .word   0
 931         .word   160
 932         .word   262144
 933         .word   16384
 934         .word   0
 935         .word   160
 936         .word   262144
 937         .word   160
 938         .word   0
 939         .word   -2147467264
 940         .word   262160
 941         .word   -2147467264
 942         .word   262160
 943         .word   -2147467104
 944         .word   16
 945         .word   -2147467104
 946         .word   16
 947         .word   -2147483648
 948         .word   16
 949         .word   -2147483488
 950         .word   16
 951         .word   16384
 952         .word   0
 953         .word   16384
 954         .word   262144
 955         .word   160
 956         .word   0
 957         .word   16544
 958         .word   262144
 959         .word   -2147483648
 960         .word   262160
 961         .word   -2147483488
 962         .word   262160
 963         .word   16544
 964         .word   262144
 965         .word   -2147483648
 966         .word   262160
 967         .word   -2147467264
 968         .word   16
 969         .word   -2147467104
 970         .word   262160
 971         .word   16544
 972         .word   0
 973         .word   160
 974         .word   262144
 975         .word   0
 976         .word   0
 977         .word   -2147483648
 978         .word   16
 979         .word   -2147467104
```



```
 980         .word   262160
 981         .word   0
 982         .word   0
 983         .word   -2147483488
 984         .word   262160
 985         .word   16544
 986         .word   0
 987         .word   0
 988         .word   262144
 989         .word   -2147467264
 990         .word   16
 991         .word   16384
 992         .word   262144
 993         .word   0
 994         .word   262144
 995         .word   -2147483488
 996         .word   16
 997         .word   67174400
 998         .word   67635200
 999         .word   0
1000         .word   67633152
1001         .word   4
1002         .word   0
1003         .word   67174404
1004         .word   67635200
1005         .word   67174400
1006         .word   0
1007         .word   67174400
1008         .word   67635200
1009         .word   0
1010         .word   2048
1011         .word   67174400
1012         .word   0
1013         .word   4
1014         .word   2048
1015         .word   67174404
1016         .word   0
1017         .word   67174404
1018         .word   67635200
1019         .word   4
1020         .word   67633152
1021         .word   67174404
1022         .word   67633152
1023         .word   4
1024         .word   67635200
1025         .word   0
1026         .word   67633152
1027         .word   0
1028         .word   2048
1029         .word   67174404
1030         .word   0
1031         .word   67174400
1032         .word   2048
1033         .word   67174400
1034         .word   67633152
1035         .word   0
1036         .word   67635200
1037         .word   4
1038         .word   67633152
1039         .word   4
1040         .word   2048
1041         .word   67174404
1042         .word   2048
1043         .word   67174404
1044         .word   67633152
1045         .word   0
```

```
1046         .word   67635200
1047         .word   0
1048         .word   0
1049         .word   0
1050         .word   0
1051         .word   67174404
1052         .word   2048
1053         .word   67174400
1054         .word   2048
1055         .word   67174400
1056         .word   67633152
1057         .word   4
1058         .word   67635200
1059         .word   4
1060         .word   0
1061         .word   4
1062         .word   67635200
1063         .word   4
1064         .word   0
1065         .word   67174404
1066         .word   67633152
1067         .word   0
1068         .word   67633152
1069         .word   0
1070         .word   2048
1071         .word   67174404
1072         .word   2048
1073         .word   0
1074         .word   67633152
1075         .word   4
1076         .word   67635200
1077         .word   67174400
1078         .word   67633152
1079         .word   0
1080         .word   2048
1081         .word   67174400
1082         .word   2048
1083         .word   67174404
1084         .word   0
1085         .word   67174404
1086         .word   2048
1087         .word   67174400
1088         .word   0
1089         .word   4
1090         .word   0
1091         .word   67174400
1092         .word   67635200
1093         .word   0
1094         .word   0
1095         .word   67174404
1096         .word   67635200
1097         .word   4
1098         .word   2048
1099         .word   67174400
1100         .word   2048
1101         .word   67174404
1102         .word   0
1103         .word   67174400
1104         .word   67633152
1105         .word   67174400
1106         .word   67635200
1107         .word   0
1108         .word   0
1109         .word   67174404
1110         .word   67635200
1111         .word   4
```



```
1112         .word   67633152
1113         .word   4
1114         .word   67633152
1115         .word   0
1116         .word   67635200
1117         .word   0
1118         .word   67635200
1119         .word   4
1120         .word   2048
1121         .word   67174400
1122         .word   0
1123         .word   67174404
1124         .word   67633152
1125         .type   des_sbox_table,#object
1126         .size   des_sbox_table,4096

1128         .align 32
1129 !
1130 ! CONSTANT POOL
1131 !
1132         .section        ".text",#alloc,#execinstr

1134 des_ip_table:
1135         .word   0
1136         .word   0
1137         .word   0
1138         .word   1024
1139         .word   8388608
1140         .word   640
1141         .word   8388608
1142         .word   1664
1143         .word   0
1144         .word   4194304
1145         .word   0
1146         .word   4195328
1147         .word   8388608
1148         .word   4194944
1149         .word   8388608
1150         .word   4195968
1151         .word   0
1152         .word   2621440
1153         .word   0
1154         .word   2622464
1155         .word   8388608
1156         .word   2622080
1157         .word   8388608
1158         .word   2623104
1159         .word   0
1160         .word   6815744
1161         .word   0
1162         .word   6816768
1163         .word   8388608
1164         .word   6816384
1165         .word   8388608
1166         .word   6817408
1167         .word   4
1168         .word   0
1169         .word   4
1170         .word   1024
1171         .word   8388612
1172         .word   640
1173         .word   8388612
1174         .word   1664
1175         .word   4
1176         .word   4194304
1177         .word   4
```

```
1178         .word   4195328
1179         .word   8388612
1180         .word   4194944
1181         .word   8388612
1182         .word   4195968
1183         .word   4
1184         .word   2621440
1185         .word   4
1186         .word   2622464
1187         .word   8388612
1188         .word   2622080
1189         .word   8388612
1190         .word   2623104
1191         .word   4
1192         .word   6815744
1193         .word   4
1194         .word   6816768
1195         .word   8388612
1196         .word   6816384
1197         .word   8388612
1198         .word   6817408
1199         .word   2
1200         .word   -2147483648
1201         .word   2
1202         .word   -2147482624
1203         .word   8388610
1204         .word   -2147483008
1205         .word   8388610
1206         .word   -2147481984
1207         .word   2
1208         .word   -2143289344
1209         .word   2
1210         .word   -2143288320
1211         .word   8388610
1212         .word   -2143288704
1213         .word   8388610
1214         .word   -2143287680
1215         .word   2
1216         .word   -2144862208
1217         .word   2
1218         .word   -2144861184
1219         .word   8388610
1220         .word   -2144861568
1221         .word   8388610
1222         .word   -2144860544
1223         .word   2
1224         .word   -2140667904
1225         .word   2
1226         .word   -2140666880
1227         .word   8388610
1228         .word   -2140667264
1229         .word   8388610
1230         .word   -2140666240
1231         .word   6
1232         .word   -2147483648
1233         .word   6
1234         .word   -2147482624
1235         .word   8388614
1236         .word   -2147483008
1237         .word   8388614
1238         .word   -2147481984
1239         .word   6
1240         .word   -2143289344
1241         .word   6
1242         .word   -2143288320
1243         .word   8388614
```



```
1244         .word   -2143288704
1245         .word   8388614
1246         .word   -2143287680
1247         .word   6
1248         .word   -2144862208
1249         .word   6
1250         .word   -2144861184
1251         .word   8388614
1252         .word   -2144861568
1253         .word   8388614
1254         .word   -2144860544
1255         .word   6
1256         .word   -2140667904
1257         .word   6
1258         .word   -2140666880
1259         .word   8388614
1260         .word   -2140667264
1261         .word   8388614
1262         .word   -2140666240
1263         .word   16384
1264         .word   0
1265         .word   16384
1266         .word   1024
1267         .word   8404992
1268         .word   640
1269         .word   8404992
1270         .word   1664
1271         .word   16384
1272         .word   4194304
1273         .word   16384
1274         .word   4195328
1275         .word   8404992
1276         .word   4194944
1277         .word   8404992
1278         .word   4195968
1279         .word   16384
1280         .word   2621440
1281         .word   16384
1282         .word   2622464
1283         .word   8404992
1284         .word   2622080
1285         .word   8404992
1286         .word   2623104
1287         .word   16384
1288         .word   6815744
1289         .word   16384
1290         .word   6816768
1291         .word   8404992
1292         .word   6816384
1293         .word   8404992
1294         .word   6817408
1295         .word   16388
1296         .word   0
1297         .word   16388
1298         .word   1024
1299         .word   8404996
1300         .word   640
1301         .word   8404996
1302         .word   1664
1303         .word   16388
1304         .word   4194304
1305         .word   16388
1306         .word   4195328
1307         .word   8404996
1308         .word   4194944
1309         .word   8404996
```

```
1310         .word   4195968
1311         .word   16388
1312         .word   2621440
1313         .word   16388
1314         .word   2622464
1315         .word   8404996
1316         .word   2622080
1317         .word   8404996
1318         .word   2623104
1319         .word   16388
1320         .word   6815744
1321         .word   16388
1322         .word   6816768
1323         .word   8404996
1324         .word   6816384
1325         .word   8404996
1326         .word   6817408
1327         .word   16386
1328         .word   -2147483648
1329         .word   16386
1330         .word   -2147482624
1331         .word   8404994
1332         .word   -2147483008
1333         .word   8404994
1334         .word   -2147481984
1335         .word   16386
1336         .word   -2143289344
1337         .word   16386
1338         .word   -2143288320
1339         .word   8404994
1340         .word   -2143288704
1341         .word   8404994
1342         .word   -2143287680
1343         .word   16386
1344         .word   -2144862208
1345         .word   16386
1346         .word   -2144861184
1347         .word   8404994
1348         .word   -2144861568
1349         .word   8404994
1350         .word   -2144860544
1351         .word   16386
1352         .word   -2140667904
1353         .word   16386
1354         .word   -2140666880
1355         .word   8404994
1356         .word   -2140667264
1357         .word   8404994
1358         .word   -2140666240
1359         .word   16390
1360         .word   -2147483648
1361         .word   16390
1362         .word   -2147482624
1363         .word   8404998
1364         .word   -2147483008
1365         .word   8404998
1366         .word   -2147481984
1367         .word   16390
1368         .word   -2143289344
1369         .word   16390
1370         .word   -2143288320
1371         .word   8404998
1372         .word   -2143288704
1373         .word   8404998
1374         .word   -2143287680
1375         .word   16390
```



```
1376         .word   -2144862208
1377         .word   16390
1378         .word   -2144861184
1379         .word   8404998
1380         .word   -2144861568
1381         .word   8404998
1382         .word   -2144860544
1383         .word   16390
1384         .word   -2140667904
1385         .word   16390
1386         .word   -2140666880
1387         .word   8404998
1388         .word   -2140667264
1389         .word   8404998
1390         .word   -2140666240
1391         .word   10240
1392         .word   0
1393         .word   10240
1394         .word   1024
1395         .word   8398848
1396         .word   640
1397         .word   8398848
1398         .word   1664
1399         .word   10240
1400         .word   4194304
1401         .word   10240
1402         .word   4195328
1403         .word   8398848
1404         .word   4194944
1405         .word   8398848
1406         .word   4195968
1407         .word   10240
1408         .word   2621440
1409         .word   10240
1410         .word   2622464
1411         .word   8398848
1412         .word   2622080
1413         .word   8398848
1414         .word   2623104
1415         .word   10240
1416         .word   6815744
1417         .word   10240
1418         .word   6816768
1419         .word   8398848
1420         .word   6816384
1421         .word   8398848
1422         .word   6817408
1423         .word   10244
1424         .word   0
1425         .word   10244
1426         .word   1024
1427         .word   8398852
1428         .word   640
1429         .word   8398852
1430         .word   1664
1431         .word   10244
1432         .word   4194304
1433         .word   10244
1434         .word   4195328
1435         .word   8398852
1436         .word   4194944
1437         .word   8398852
1438         .word   4195968
1439         .word   10244
1440         .word   2621440
1441         .word   10244
```

```
1442         .word   2622464
1443         .word   8398852
1444         .word   2622080
1445         .word   8398852
1446         .word   2623104
1447         .word   10244
1448         .word   6815744
1449         .word   10244
1450         .word   6816768
1451         .word   8398852
1452         .word   6816384
1453         .word   8398852
1454         .word   6817408
1455         .word   10242
1456         .word   -2147483648
1457         .word   10242
1458         .word   -2147482624
1459         .word   8398850
1460         .word   -2147483008
1461         .word   8398850
1462         .word   -2147481984
1463         .word   10242
1464         .word   -2143289344
1465         .word   10242
1466         .word   -2143288320
1467         .word   8398850
1468         .word   -2143288704
1469         .word   8398850
1470         .word   -2143287680
1471         .word   10242
1472         .word   -2144862208
1473         .word   10242
1474         .word   -2144861184
1475         .word   8398850
1476         .word   -2144861568
1477         .word   8398850
1478         .word   -2144860544
1479         .word   10242
1480         .word   -2140667904
1481         .word   10242
1482         .word   -2140666880
1483         .word   8398850
1484         .word   -2140667264
1485         .word   8398850
1486         .word   -2140666240
1487         .word   10246
1488         .word   -2147483648
1489         .word   10246
1490         .word   -2147482624
1491         .word   8398854
1492         .word   -2147483008
1493         .word   8398854
1494         .word   -2147481984
1495         .word   10246
1496         .word   -2143289344
1497         .word   10246
1498         .word   -2143288320
1499         .word   8398854
1500         .word   -2143288704
1501         .word   8398854
1502         .word   -2143287680
1503         .word   10246
1504         .word   -2144862208
1505         .word   10246
1506         .word   -2144861184
1507         .word   8398854
```



```
1508         .word   -2144861568
1509         .word   8398854
1510         .word   -2144860544
1511         .word   10246
1512         .word   -2140667904
1513         .word   10246
1514         .word   -2140666880
1515         .word   8398854
1516         .word   -2140667264
1517         .word   8398854
1518         .word   -2140666240
1519         .word   26624
1520         .word   0
1521         .word   26624
1522         .word   1024
1523         .word   8415232
1524         .word   640
1525         .word   8415232
1526         .word   1664
1527         .word   26624
1528         .word   4194304
1529         .word   26624
1530         .word   4195328
1531         .word   8415232
1532         .word   4194944
1533         .word   8415232
1534         .word   4195968
1535         .word   26624
1536         .word   2621440
1537         .word   26624
1538         .word   2622464
1539         .word   8415232
1540         .word   2622080
1541         .word   8415232
1542         .word   2623104
1543         .word   26624
1544         .word   6815744
1545         .word   26624
1546         .word   6816768
1547         .word   8415232
1548         .word   6816384
1549         .word   8415232
1550         .word   6817408
1551         .word   26628
1552         .word   0
1553         .word   26628
1554         .word   1024
1555         .word   8415236
1556         .word   640
1557         .word   8415236
1558         .word   1664
1559         .word   26628
1560         .word   4194304
1561         .word   26628
1562         .word   4195328
1563         .word   8415236
1564         .word   4194944
1565         .word   8415236
1566         .word   4195968
1567         .word   26628
1568         .word   2621440
1569         .word   26628
1570         .word   2622464
1571         .word   8415236
1572         .word   2622080
1573         .word   8415236
```

```
1574         .word   2623104
1575         .word   26628
1576         .word   6815744
1577         .word   26628
1578         .word   6816768
1579         .word   8415236
1580         .word   6816384
1581         .word   8415236
1582         .word   6817408
1583         .word   26626
1584         .word   -2147483648
1585         .word   26626
1586         .word   -2147482624
1587         .word   8415234
1588         .word   -2147483008
1589         .word   8415234
1590         .word   -2147481984
1591         .word   26626
1592         .word   -2143289344
1593         .word   26626
1594         .word   -2143288320
1595         .word   8415234
1596         .word   -2143288704
1597         .word   8415234
1598         .word   -2143287680
1599         .word   26626
1600         .word   -2144862208
1601         .word   26626
1602         .word   -2144861184
1603         .word   8415234
1604         .word   -2144861568
1605         .word   8415234
1606         .word   -2144860544
1607         .word   26626
1608         .word   -2140667904
1609         .word   26626
1610         .word   -2140666880
1611         .word   8415234
1612         .word   -2140667264
1613         .word   8415234
1614         .word   -2140666240
1615         .word   26630
1616         .word   -2147483648
1617         .word   26630
1618         .word   -2147482624
1619         .word   8415238
1620         .word   -2147483008
1621         .word   8415238
1622         .word   -2147481984
1623         .word   26630
1624         .word   -2143289344
1625         .word   26630
1626         .word   -2143288320
1627         .word   8415238
1628         .word   -2143288704
1629         .word   8415238
1630         .word   -2143287680
1631         .word   26630
1632         .word   -2144862208
1633         .word   26630
1634         .word   -2144861184
1635         .word   8415238
1636         .word   -2144861568
1637         .word   8415238
1638         .word   -2144860544
1639         .word   26630
```



```
1640         .word   -2140667904
1641         .word   26630
1642         .word   -2140666880
1643         .word   8415238
1644         .word   -2140667264
1645         .word   8415238
1646         .word   -2140666240
1647         .word   0
1648         .word   0
1649         .word   0
1650         .word   20480
1651         .word   0
1652         .word   2048
1653         .word   0
1654         .word   22528
1655         .word   0
1656         .word   83886080
1657         .word   0
1658         .word   83906560
1659         .word   0
1660         .word   83888128
1661         .word   0
1662         .word   83908608
1663         .word   0
1664         .word   8388608
1665         .word   0
1666         .word   8409088
1667         .word   0
1668         .word   8390656
1669         .word   0
1670         .word   8411136
1671         .word   0
1672         .word   92274688
1673         .word   0
1674         .word   92295168
1675         .word   0
1676         .word   92276736
1677         .word   0
1678         .word   92297216
1679         .word   80
1680         .word   0
1681         .word   80
1682         .word   20480
1683         .word   80
1684         .word   2048
1685         .word   80
1686         .word   22528
1687         .word   80
1688         .word   83886080
1689         .word   80
1690         .word   83906560
1691         .word   80
1692         .word   83888128
1693         .word   80
1694         .word   83908608
1695         .word   80
1696         .word   8388608
1697         .word   80
1698         .word   8409088
1699         .word   80
1700         .word   8390656
1701         .word   80
1702         .word   8411136
1703         .word   80
1704         .word   92274688
1705         .word   80
```

```
1706          .word   92295168
1707          .word   80
1708          .word   92276736
1709          .word   80
1710          .word   92297216
1711          .word   8
1712          .word   0
1713          .word   8
1714          .word   20480
1715          .word   8
1716          .word   2048
1717          .word   8
1718          .word   22528
1719          .word   8
1720          .word   83886080
1721          .word   8
1722          .word   83906560
1723          .word   8
1724          .word   83888128
1725          .word   8
1726          .word   83908608
1727          .word   8
1728          .word   8388608
1729          .word   8
1730          .word   8409088
1731          .word   8
1732          .word   8390656
1733          .word   8
1734          .word   8411136
1735          .word   8
1736          .word   92274688
1737          .word   8
1738          .word   92295168
1739          .word   8
1740          .word   92276736
1741          .word   8
1742          .word   92297216
1743          .word   88
1744          .word   0
1745          .word   88
1746          .word   20480
1747          .word   88
1748          .word   2048
1749          .word   88
1750          .word   22528
1751          .word   88
1752          .word   83886080
1753          .word   88
1754          .word   83906560
1755          .word   88
1756          .word   83888128
1757          .word   88
1758          .word   83908608
1759          .word   88
1760          .word   8388608
1761          .word   88
1762          .word   8409088
1763          .word   88
1764          .word   8390656
1765          .word   88
1766          .word   8411136
1767          .word   88
1768          .word   92274688
1769          .word   88
1770          .word   92295168
1771          .word   88
```



```
1772          .word   92276736
1773          .word   88
1774          .word   92297216
1775          .word   327680
1776          .word   4
1777          .word   327680
1778          .word   20484
1779          .word   327680
1780          .word   2052
1781          .word   327680
1782          .word   22532
1783          .word   327680
1784          .word   83886084
1785          .word   327680
1786          .word   83906564
1787          .word   327680
1788          .word   83888132
1789          .word   327680
1790          .word   83908612
1791          .word   327680
1792          .word   8388612
1793          .word   327680
1794          .word   8409092
1795          .word   327680
1796          .word   8390660
1797          .word   327680
1798          .word   8411140
1799          .word   327680
1800          .word   92274692
1801          .word   327680
1802          .word   92295172
1803          .word   327680
1804          .word   92276740
1805          .word   327680
1806          .word   92297220
1807          .word   327760
1808          .word   4
1809          .word   327760
1810          .word   20484
1811          .word   327760
1812          .word   2052
1813          .word   327760
1814          .word   22532
1815          .word   327760
1816          .word   83886084
1817          .word   327760
1818          .word   83906564
1819          .word   327760
1820          .word   83888132
1821          .word   327760
1822          .word   83908612
1823          .word   327760
1824          .word   8388612
1825          .word   327760
1826          .word   8409092
1827          .word   327760
1828          .word   8390660
1829          .word   327760
1830          .word   8411140
1831          .word   327760
1832          .word   92274692
1833          .word   327760
1834          .word   92295172
1835          .word   327760
1836          .word   92276740
1837          .word   327760
```

```
1838         .word   92297220
1839         .word   327688
1840         .word   4
1841         .word   327688
1842         .word   20484
1843         .word   327688
1844         .word   2052
1845         .word   327688
1846         .word   22532
1847         .word   327688
1848         .word   83886084
1849         .word   327688
1850         .word   83906564
1851         .word   327688
1852         .word   83888132
1853         .word   327688
1854         .word   83908612
1855         .word   327688
1856         .word   8388612
1857         .word   327688
1858         .word   8409092
1859         .word   327688
1860         .word   8390660
1861         .word   327688
1862         .word   8411140
1863         .word   327688
1864         .word   92274692
1865         .word   327688
1866         .word   92295172
1867         .word   327688
1868         .word   92276740
1869         .word   327688
1870         .word   92297220
1871         .word   327768
1872         .word   4
1873         .word   327768
1874         .word   20484
1875         .word   327768
1876         .word   2052
1877         .word   327768
1878         .word   22532
1879         .word   327768
1880         .word   83886084
1881         .word   327768
1882         .word   83906564
1883         .word   327768
1884         .word   83888132
1885         .word   327768
1886         .word   83908612
1887         .word   327768
1888         .word   8388612
1889         .word   327768
1890         .word   8409092
1891         .word   327768
1892         .word   8390660
1893         .word   327768
1894         .word   8411140
1895         .word   327768
1896         .word   92274692
1897         .word   327768
1898         .word   92295172
1899         .word   327768
1900         .word   92276740
1901         .word   327768
1902         .word   92297220
1903         .word   32768
```



```
1904         .word   0
1905         .word   32768
1906         .word   20480
1907         .word   32768
1908         .word   2048
1909         .word   32768
1910         .word   22528
1911         .word   32768
1912         .word   83886080
1913         .word   32768
1914         .word   83906560
1915         .word   32768
1916         .word   83888128
1917         .word   32768
1918         .word   83908608
1919         .word   32768
1920         .word   8388608
1921         .word   32768
1922         .word   8409088
1923         .word   32768
1924         .word   8390656
1925         .word   32768
1926         .word   8411136
1927         .word   32768
1928         .word   92274688
1929         .word   32768
1930         .word   92295168
1931         .word   32768
1932         .word   92276736
1933         .word   32768
1934         .word   92297216
1935         .word   32848
1936         .word   0
1937         .word   32848
1938         .word   20480
1939         .word   32848
1940         .word   2048
1941         .word   32848
1942         .word   22528
1943         .word   32848
1944         .word   83886080
1945         .word   32848
1946         .word   83906560
1947         .word   32848
1948         .word   83888128
1949         .word   32848
1950         .word   83908608
1951         .word   32848
1952         .word   8388608
1953         .word   32848
1954         .word   8409088
1955         .word   32848
1956         .word   8390656
1957         .word   32848
1958         .word   8411136
1959         .word   32848
1960         .word   92274688
1961         .word   32848
1962         .word   92295168
1963         .word   32848
1964         .word   92276736
1965         .word   32848
1966         .word   92297216
1967         .word   32776
1968         .word   0
1969         .word   32776
```

```
1970         .word   20480
1971         .word   32776
1972         .word   2048
1973         .word   32776
1974         .word   22528
1975         .word   32776
1976         .word   83886080
1977         .word   32776
1978         .word   83906560
1979         .word   32776
1980         .word   83888128
1981         .word   32776
1982         .word   83908608
1983         .word   32776
1984         .word   8388608
1985         .word   32776
1986         .word   8409088
1987         .word   32776
1988         .word   8390656
1989         .word   32776
1990         .word   8411136
1991         .word   32776
1992         .word   92274688
1993         .word   32776
1994         .word   92295168
1995         .word   32776
1996         .word   92276736
1997         .word   32776
1998         .word   92297216
1999         .word   32856
2000         .word   0
2001         .word   32856
2002         .word   20480
2003         .word   32856
2004         .word   2048
2005         .word   32856
2006         .word   22528
2007         .word   32856
2008         .word   83886080
2009         .word   32856
2010         .word   83906560
2011         .word   32856
2012         .word   83888128
2013         .word   32856
2014         .word   83908608
2015         .word   32856
2016         .word   8388608
2017         .word   32856
2018         .word   8409088
2019         .word   32856
2020         .word   8390656
2021         .word   32856
2022         .word   8411136
2023         .word   32856
2024         .word   92274688
2025         .word   32856
2026         .word   92295168
2027         .word   32856
2028         .word   92276736
2029         .word   32856
2030         .word   92297216
2031         .word   360448
2032         .word   4
2033         .word   360448
2034         .word   20484
2035         .word   360448
```



```
2036         .word   2052
2037         .word   360448
2038         .word   22532
2039         .word   360448
2040         .word   83886084
2041         .word   360448
2042         .word   83906564
2043         .word   360448
2044         .word   83888132
2045         .word   360448
2046         .word   83908612
2047         .word   360448
2048         .word   8388612
2049         .word   360448
2050         .word   8409092
2051         .word   360448
2052         .word   8390660
2053         .word   360448
2054         .word   8411140
2055         .word   360448
2056         .word   92274692
2057         .word   360448
2058         .word   92295172
2059         .word   360448
2060         .word   92276740
2061         .word   360448
2062         .word   92297220
2063         .word   360528
2064         .word   4
2065         .word   360528
2066         .word   20484
2067         .word   360528
2068         .word   2052
2069         .word   360528
2070         .word   22532
2071         .word   360528
2072         .word   83886084
2073         .word   360528
2074         .word   83906564
2075         .word   360528
2076         .word   83888132
2077         .word   360528
2078         .word   83908612
2079         .word   360528
2080         .word   8388612
2081         .word   360528
2082         .word   8409092
2083         .word   360528
2084         .word   8390660
2085         .word   360528
2086         .word   8411140
2087         .word   360528
2088         .word   92274692
2089         .word   360528
2090         .word   92295172
2091         .word   360528
2092         .word   92276740
2093         .word   360528
2094         .word   92297220
2095         .word   360456
2096         .word   4
2097         .word   360456
2098         .word   20484
2099         .word   360456
2100         .word   2052
2101         .word   360456
```

```
2102         .word   22532
2103         .word   360456
2104         .word   83886084
2105         .word   360456
2106         .word   83906564
2107         .word   360456
2108         .word   83888132
2109         .word   360456
2110         .word   83908612
2111         .word   360456
2112         .word   8388612
2113         .word   360456
2114         .word   8409092
2115         .word   360456
2116         .word   8390660
2117         .word   360456
2118         .word   8411140
2119         .word   360456
2120         .word   92274692
2121         .word   360456
2122         .word   92295172
2123         .word   360456
2124         .word   92276740
2125         .word   360456
2126         .word   92297220
2127         .word   360536
2128         .word   4
2129         .word   360536
2130         .word   20484
2131         .word   360536
2132         .word   2052
2133         .word   360536
2134         .word   22532
2135         .word   360536
2136         .word   83886084
2137         .word   360536
2138         .word   83906564
2139         .word   360536
2140         .word   83888132
2141         .word   360536
2142         .word   83908612
2143         .word   360536
2144         .word   8388612
2145         .word   360536
2146         .word   8409092
2147         .word   360536
2148         .word   8390660
2149         .word   360536
2150         .word   8411140
2151         .word   360536
2152         .word   92274692
2153         .word   360536
2154         .word   92295172
2155         .word   360536
2156         .word   92276740
2157         .word   360536
2158         .word   92297220
2159         .type   des_ip_table,#object
2160         .size   des_ip_table,4096


2163         .section        ".data",#alloc
2164                 .align  32


2167 des_enc_const:
```



```
2169 #ifdef __sparcv9

2171 !
2172 ! For v9, the addresses ar 64-bit long, so we should use .xword
2173 ! instead of .word, this makes the constant table bigger
2174 !
2175         .xword  (des_ip_table+0x0)      ! initial permutation table
2176         .xword  (des_ip_table+0x800)
2177
2178         .xword  (des_fp_table+0x0)      ! final permutation table
2179
2180         .xword  (des_sbox_table+0x0)    ! sboxes table
2181         .xword  (des_sbox_table+0x200)
2182         .xword  (des_sbox_table+0x400)
2183         .xword  (des_sbox_table+0x600)
2184         .xword  (des_sbox_table+0x800)
2185         .xword  (des_sbox_table+0xa00)
2186         .xword  (des_sbox_table+0xc00)
2187         .xword  (des_sbox_table+0xe00)

2189         .word   0                       ! for alignment
2190         .word   7                       ! counter for encrypt loop
2191
2192         .word   16515072                ! top_1
2193         .word   0                       !
2194
2195         .word   262143                  ! mid_4
2196         .word   -67108864               !
2197
2198         .word   0                       ! low_3
2199         .word   67108608                !
2200
2201         .word   -1431655766             ! 0xaaaaaaaaaaaaaaaa
2202         .word   -1431655766             !
2203
2204         .word   1431655765              ! 0x5555555555555555
2205         .word   1431655765              !
2206 #else

2208 !
2209 ! For v8, the addresses are 32-bit long
2210 !

2212         .word   (des_ip_table+0x0)      ! initial permutation table
2213         .word   (des_ip_table+0x800)
2214
2215         .word   (des_fp_table+0x0)      ! final permutation table
2216
2217         .word   (des_sbox_table+0x0)    ! sboxes table
2218         .word   (des_sbox_table+0x200)
2219         .word   (des_sbox_table+0x400)
2220         .word   (des_sbox_table+0x600)
2221         .word   (des_sbox_table+0x800)
2222         .word   (des_sbox_table+0xa00)
2223         .word   (des_sbox_table+0xc00)
2224         .word   (des_sbox_table+0xe00)
2225
2226         .word   7                       ! counter for encrypt loop
2227
2228         .word   16515072                ! top_1
2229         .word   0                       !
2230
2231         .word   262143                  ! mid_4
2232         .word   -67108864               !
2233
```

```
2234         .word   0                       ! low_3
2235         .word   67108608                !
2236
2237         .word   -1431655766             ! 0xaaaaaaaaaaaaaaaa
2238         .word   -1431655766             !
2239
2240         .word   1431655765              ! 0x5555555555555555
2241         .word   1431655765              !
2242 #endif
2243         .type   des_enc_const,#object
2244         .size   des_enc_const,(.-des_enc_const)


2247         .section        ".text",#alloc,#execinstr
2248 /* 000000          0 */         .align  32
2249 /* 000000            */         .skip   32
2250 !
2251 ! SUBROUTINE des_crypt_impl
2252 !
2253 ! OFFSET    SOURCE LINE LABEL   INSTRUCTION

2255 .global des_crypt_impl

2257 ! uint64_t des_crypt_impl(uint64_t *ks, uint64_t block, int one_or_three);
2258 !
2259 ! ks is the key schedule, en/decryption is differentiated by computing
2260 !    an encryption key schedule for encryption and the reverse of it
2261 !    for decryption (for DES, 16 entries, for triple-DES, 48 entries)
2262 ! block is the 64-bit block to en/decrypt
2263 ! one_or_three is 1 for DES and 3 for triple-DES
2264
2265         des_crypt_impl:

2267 #ifdef __sparcv9
2268         save    %sp,-192,%sp
2269 #ifdef PIC
2270 .L0:
2271         call    . + 8
2272         sethi   %hi(_GLOBAL_OFFSET_TABLE_ - (.L0 - .)), %o1
2273         sethi   %hi(des_enc_const), %g1
2274
2275         or      %o1, %lo(_GLOBAL_OFFSET_TABLE_ - (.L0 - .)), %o1
2276         or      %g1, %lo(des_enc_const),%g1

2278         add     %o1, %o7, %o1
2279 #else
2280         sethi   %hh(des_enc_const),%o1
2281         sethi   %lm(des_enc_const),%g1

2283         or      %o1,%hm(des_enc_const),%o1
2284         or      %g1,%lo(des_enc_const),%g1

2286         sllx    %o1,32,%o1
2287 #endif
2288         sethi   %hi(0xaaaaaaaa), %g3

2290 #ifdef PIC
2291         ldx     [%o1 + %g1], %i5
2292 #else
2293         or      %o1,%g1,%i5             ! &des_enc_const
2294 #endif
2295         or      %g3, %lo(0xaaaaaaaa), %g3

2297         sllx    %g3, 32, %o0

2299         or      %g3, %o0, %g3           ! 0xaaaaaaaaaaaaaaaa
```



```
2301         srlx    %g3, 1, %g2             ! 0x5555555555555555
2302         and     %i1, %g3, %g1

2304         sllx    %g1, 7, %g3
2305         ldx     [%i5 + 0], %l7          ! &(des_ip_table[0][0])
2306         and     %i1, %g2, %g2

2308         srlx    %g2, 7, %g4
2309         ldx     [%i5 + 8], %l6          ! &(des_ip_table[1][0])
2310         or      %g1, %g3, %g1

2312         srlx    %g1, 21, %o0
2313         ld      [%i5 + 92], %i4         ! 7 (for iteration counter)
2314         or      %g2, %g4, %g2

2316         srlx    %g1, 5, %o1
2317         ldx     [%i5 + 24], %l0         ! &(des_sbox_table[0][0])
2318         and     %o0, 0x7f8, %o0

2320         srlx    %g2, 13, %o2
2321         ldx     [%l7 + %o0], %o0
2322         and     %o1, 0x7f8, %o1

2324         sllx    %g2, 3, %o3
2325         ldx     [%l6 + %o1], %o1
2326         and     %o2, 0x7f8, %o2

2328         srlx    %g1, 53, %o4
2329         ldx     [%l7 + %o2], %o2
2330         and     %o3, 0x7f8, %o3

2332         srlx    %g1, 37, %o5
2333         ldx     [%l6 + %o3], %o3
2334         and     %o4, 0x7f8, %o4

2336         srlx    %g2, 45, %g1
2337         ldx     [%l7 + %o4], %o4
2338         and     %o5, 0x7f8, %o5
2339
2340         srlx    %g2, 29, %g2
2341         ldx     [%l6 + %o5], %o5
2342         and     %g1, 0x7f8, %g1

2344         sllx    %o0, 6, %o0
2345         ldx     [%l7 + %g1], %g1
2346         and     %g2, 0x7f8, %g2

2348         sllx    %o1, 6, %o1
2349         ldx     [%l6 + %g2], %g2
2350         or      %o4, %o5, %o4

2352         sllx    %o2, 6, %o2
2353         ldx     [%i5 + 32], %l1         ! &(des_sbox_table[1][0])
2354         or      %o0, %o1, %o0

2356         sllx    %o3, 6, %o3
2357         ldx     [%i5 + 96], %g3         ! top_1
2358         or      %o0, %o4, %o0

2360         or      %g1, %g2, %g1
2361         ldx     [%i5 + 104], %g4        ! mid_4
2362         or      %o2, %o3, %o2

2364         and     %o0, %g3, %o4
2365         ldx     [%i5 + 112], %g2        ! low_3
```

```
2366         or      %o2, %g1, %o1

2368         sllx    %o4, 8, %o4
2369         ldx     [%i5 + 40], %l2         ! &(des_sbox_table[2][0])
2370         and     %o1, %g3, %o5

2372         sllx    %o5, 8, %o5
2373         ldx     [%i5 + 48], %l3         ! &(des_sbox_table[3][0])
2374         and     %o0, %g2, %o2

2376         srlx    %o2, 5, %o2
2377         ldx     [%i5 + 56], %l4         ! &(des_sbox_table[4][0])
2378         and     %o1, %g2, %o3

2380         srlx    %o3, 5, %o3
2381         ldx     [%i0], %g2              ! ks[0]
2382         and     %o0, %g4, %o0

2384         or      %o0, %o2, %o0
2385         ldx     [%i5 + 64], %l5         ! &(des_sbox_table[5][0])
2386         and     %o1, %g4, %o1

2388         or      %o0, %o4, %o0
2389         ldx     [%i5 + 72], %l6         ! &(des_sbox_table[6][0])
2390         or      %o1, %o3, %o1

2392         xor     %o0, %g2, %g1
2393         ldx     [%i5 + 80], %l7         ! &(des_sbox_table[7][0])
2394         or      %o1, %o5, %o1
2395 #else

2397 ! v8 version

2399         save    %sp,-144,%sp
2400         sethi   %hi(des_enc_const), %g2

2402 #ifdef PIC
2403 .L1:
2404         call    .+8

2406         sethi   %hi(_GLOBAL_OFFSET_TABLE_ - (.L1 - .)), %g1

2408         or      %g1, %lo(_GLOBAL_OFFSET_TABLE_ - (.L1 - .)), %g1
2409         or      %g2, %lo(des_enc_const), %g2
2410 #else
2411         or      %g2, %lo(des_enc_const), %i5
2412 #endif
2413         srl     %i2, 0, %g4
2414         sethi   %hi(0xaaaaaaaa), %g3

2416         sllx    %i1, 32, %g5
2417         or      %g3, %lo(0xaaaaaaaa), %g3

2419         sllx    %g3, 32, %o0
2420         add     %o7, %g1, %g1

2422         or      %g3, %o0, %g3           ! 0xaaaaaaaaaaaaaaaa
2423 #ifdef PIC
2424         ld      [%g1 + %g2], %i5
2425 #endif
2426         or      %g4, %g5, %g4

2428         srlx    %g3, 1, %g2             ! 0x5555555555555555
2429         and     %g4, %g3, %g1

2431         sllx    %g1, 7, %g3
```



```
2432         ld      [%i5 + 0], %l7          ! &(des_ip_table[0][0])
2433         and     %g4, %g2, %g2

2435         srlx    %g2, 7, %g4
2436         ld      [%i5 + 4], %l6          ! &(des_ip_table[1][0])
2437         or      %g1, %g3, %g1

2439         srlx    %g1, 21, %o0
2440         ld      [%i5 + 44], %i4         ! 7 (for iteration counter)
2441         or      %g2, %g4, %g2

2443         srlx    %g1, 5, %o1
2444         ld      [%i5 + 12], %l0         ! &(des_sbox_table[0][0])
2445         and     %o0, 0x7f8, %o0

2447         srlx    %g2, 13, %o2
2448         ldx     [%l7 + %o0], %o0
2449         and     %o1, 0x7f8, %o1

2451         sllx    %g2, 3, %o3
2452         ldx     [%l6 + %o1], %o1
2453         and     %o2, 0x7f8, %o2

2455         srlx    %g1, 53, %o4
2456         ldx     [%l7 + %o2], %o2
2457         and     %o3, 0x7f8, %o3

2459         srlx    %g1, 37, %o5
2460         ldx     [%l6 + %o3], %o3
2461         and     %o4, 0x7f8, %o4

2463         srlx    %g2, 45, %g1
2464         ldx     [%l7 + %o4], %o4
2465         and     %o5, 0x7f8, %o5
2466
2467         srlx    %g2, 29, %g2
2468         ldx     [%l6 + %o5], %o5
2469         and     %g1, 0x7f8, %g1

2471         sllx    %o0, 6, %o0
2472         ldx     [%l7 + %g1], %g1
2473         and     %g2, 0x7f8, %g2

2475         sllx    %o1, 6, %o1
2476         ldx     [%l6 + %g2], %g2
2477         or      %o4, %o5, %o4

2479         sllx    %o2, 6, %o2
2480         ld      [%i5 + 16], %l1         ! &(des_sbox_table[1][0])
2481         or      %o0, %o1, %o0

2483         sllx    %o3, 6, %o3
2484         ldx     [%i5 + 48], %g3         ! top_1
2485         or      %o0, %o4, %o0

2487         or      %g1, %g2, %g1
2488         ldx     [%i5 + 56], %g4         ! mid_4
2489         or      %o2, %o3, %o2

2491         and     %o0, %g3, %o4
2492         ldx     [%i5 + 64], %g2         ! low_3
2493         or      %o2, %g1, %o1

2495         sllx    %o4, 8, %o4
2496         ld      [%i5 + 20], %l2         ! &(des_sbox_table[2][0])
2497         and     %o1, %g3, %o5
```

```
2499          sllx    %o5, 8, %o5
2500          ld      [%i5 + 24], %l3         ! &(des_sbox_table[3][0])
2501          and     %o0, %g2, %o2

2503          srlx    %o2, 5, %o2
2504          ld      [%i5 + 28], %l4         ! &(des_sbox_table[4][0])
2505          and     %o1, %g2, %o3

2507          srlx    %o3, 5, %o3
2508          ldx     [%i0], %g2              ! ks[0]
2509          and     %o0, %g4, %o0

2511          or      %o0, %o2, %o0
2512          ld      [%i5 + 32], %l5         ! &(des_sbox_table[5][0])
2513          and     %o1, %g4, %o1

2515          or      %o0, %o4, %o0
2516          ld      [%i5 + 36], %l6         ! &(des_sbox_table[6][0])
2517          or      %o1, %o3, %o1

2519          xor     %o0, %g2, %g1
2520          ld      [%i5 + 40], %l7         ! &(des_sbox_table[7][0])
2521          or      %o1, %o5, %o1

2523 #endif
2524 .L2:
2525          srlx    %g1, 55, %g1
2526          xor     %o0, %g2, %o5
2527 .L3:
2528          srlx    %o5, 41, %g3
2529          ldx     [%l0 + %g1], %g1
2530          and     %o5, 0x1f8, %g2

2532          srlx    %o5, 29, %g4
2533          ldx     [%l7 + %g2], %g2
2534          and     %g3, 0x1f8, %g3

2536          srlx    %o5, 12, %o2
2537          ldx     [%l1 + %g3], %g3
2538          and     %g4, 0x1f8, %g4

2540          srlx    %o5, 35, %o3
2541          ldx     [%l3 + %g4], %g4
2542          and     %o2, 0x1f8, %o2

2544          srlx    %o5, 6, %o4
2545          ldx     [%l5 + %o2], %o2
2546          and     %o3, 0x1f8, %o3

2548          or      %g1, %g2, %g2
2549          ldx     [%l2 + %o3], %o3
2550          and     %o4, 0x1f8, %o4

2552          srl     %o5, 23, %o5
2553          ldx     [%l6 + %o4], %o4
2554          or      %g3, %g4, %g4

2556          or      %g2, %g4, %g4
2557          ldx     [%l4 + %o5], %o5
2558          or      %o2, %o3, %o3

2560          or      %g4, %o3, %o3
2561          ldx     [%i0 + 8], %g2
2562          add     %i0, 16, %i0
```



```
2564          xor     %o1, %o3, %o1
2565          or      %o4, %o5, %o5

2567          xor     %o1, %g2, %g1
2568          xor     %o1, %o5, %o1

2570          srl     %g1, 23, %g1
2571          xor     %o1, %g2, %o5

2573          srlx    %o5, 41, %g3
2574          ldx     [%l4 + %g1], %g1
2575          and     %o5, 0x1f8, %g2

2577          srlx    %o5, 29, %g4
2578          ldx     [%l7 + %g2], %g2
2579          and     %g3, 0x1f8, %g3

2581          srlx    %o5, 12, %o2
2582          ldx     [%l1 + %g3], %g3
2583          and     %g4, 0x1f8, %g4

2585          srlx    %o5, 6, %o3
2586          ldx     [%l3 + %g4], %g4
2587          and     %o2, 0x1f8, %o2

2589          srlx    %o5, 35, %o4
2590          ldx     [%l5 + %o2], %o2
2591          and     %o3, 0x1f8, %o3

2593          or      %g1, %g2, %g2
2594          ldx     [%l6 + %o3], %o3
2595          and     %o4, 0x1f8, %o4

2597          srlx    %o5, 55, %o5
2598          ldx     [%l2 + %o4], %o4
2599          or      %g3, %g4, %g4

2601          or      %g2, %g4, %g4
2602          ldx     [%l0 + %o5], %o5
2603          or      %o2, %o3, %o3

2605          or      %g4, %o3, %o3
2606          ldx     [%i0], %g2
2607          subcc   %i4,1,%i4

2609          xor     %o0, %o3, %o0
2610          or      %o4, %o5, %o5

2612          xor     %o0, %g2, %g1
2613 !        bnz     %icc, .L2
2614          xor     %o0, %o5, %o0

2616          srlx    %g1, 55, %g1
2617          xor     %o0, %g2, %o5

2619          srlx    %o5, 41, %g3
2620          ldx     [%l0 + %g1], %g1
2621          and     %o5, 0x1f8, %g2

2623          srlx    %o5, 29, %g4
2624          ldx     [%l7 + %g2], %g2
2625          and     %g3, 0x1f8, %g3

2627          srlx    %o5, 12, %o2
2628          ldx     [%l1 + %g3], %g3
2629          and     %g4, 0x1f8, %g4
```

```
2631         srlx    %o5, 35, %o3
2632         ldx     [%l3 + %g4], %g4
2633         and     %o2, 0x1f8, %o2

2635         srlx    %o5, 6, %o4
2636         ldx     [%l5 + %o2], %o2
2637         and     %o3, 0x1f8, %o3

2639         or      %g1, %g2, %g2
2640         ldx     [%l2 + %o3], %o3
2641         and     %o4, 0x1f8, %o4

2643         srl     %o5, 23, %o5
2644         ldx     [%l6 + %o4], %o4
2645         or      %g3, %g4, %g4

2647         or      %g2, %g4, %g4
2648         ldx     [%l4 + %o5], %o5
2649         or      %o2, %o3, %o3

2651         or      %g4, %o3, %o3
2652         ldx     [%i0 + 8], %g2
2653         add     %i0, 16, %i0

2655         xor     %o1, %o3, %o1
2656         or      %o4, %o5, %o5

2658         xor     %o1, %g2, %g1
2659         xor     %o1, %o5, %o1

2661         srl     %g1, 23, %g1
2662         xor     %o1, %g2, %o5

2664         srlx    %o5, 41, %g3
2665         ldx     [%l4 + %g1], %g1
2666         and     %o5, 0x1f8, %g2

2668         srlx    %o5, 29, %g4
2669         ldx     [%l7 + %g2], %g2
2670         and     %g3, 0x1f8, %g3

2672         srlx    %o5, 12, %o2
2673         ldx     [%l1 + %g3], %g3
2674         and     %g4, 0x1f8, %g4

2676         srlx    %o5, 6, %o3
2677         ldx     [%l3 + %g4], %g4
2678         and     %o2, 0x1f8, %o2

2680         srlx    %o5, 35, %o4
2681         ldx     [%l5 + %o2], %o2
2682         and     %o3, 0x1f8, %o3

2684         or      %g1, %g2, %g2
2685         ldx     [%l6 + %o3], %o3
2686         and     %o4, 0x1f8, %o4

2688         srlx    %o5, 55, %o5
2689         ldx     [%l2 + %o4], %o4
2690         or      %g3, %g4, %g4

2692         or      %g2, %g4, %g4
2693         ldx     [%l0 + %o5], %o5
2694         or      %o2, %o3, %o3
```



```
2696         or      %g4, %o3, %o3
2697         ldx     [%i0], %g2
2698         subcc   %i4,1,%i4

2700         xor     %o0, %o3, %o0
2701         or      %o4, %o5, %o5

2703         xor     %o0, %g2, %g1
2704 !       bnz     %icc, .L2
2705         xor     %o0, %o5, %o0

2707         srlx    %g1, 55, %g1
2708         xor     %o0, %g2, %o5

2710         srlx    %o5, 41, %g3
2711         ldx     [%l0 + %g1], %g1
2712         and     %o5, 0x1f8, %g2

2714         srlx    %o5, 29, %g4
2715         ldx     [%l7 + %g2], %g2
2716         and     %g3, 0x1f8, %g3

2718         srlx    %o5, 12, %o2
2719         ldx     [%l1 + %g3], %g3
2720         and     %g4, 0x1f8, %g4

2722         srlx    %o5, 35, %o3
2723         ldx     [%l3 + %g4], %g4
2724         and     %o2, 0x1f8, %o2

2726         srlx    %o5, 6, %o4
2727         ldx     [%l5 + %o2], %o2
2728         and     %o3, 0x1f8, %o3

2730         or      %g1, %g2, %g2
2731         ldx     [%l2 + %o3], %o3
2732         and     %o4, 0x1f8, %o4

2734         srl     %o5, 23, %o5
2735         ldx     [%l6 + %o4], %o4
2736         or      %g3, %g4, %g4

2738         or      %g2, %g4, %g4
2739         ldx     [%l4 + %o5], %o5
2740         or      %o2, %o3, %o3

2742         or      %g4, %o3, %o3
2743         ldx     [%i0 + 8], %g2
2744         add     %i0, 16, %i0

2746         xor     %o1, %o3, %o1
2747         or      %o4, %o5, %o5

2749         xor     %o1, %g2, %g1
2750         xor     %o1, %o5, %o1

2752         srl     %g1, 23, %g1
2753         xor     %o1, %g2, %o5

2755         srlx    %o5, 41, %g3
2756         ldx     [%l4 + %g1], %g1
2757         and     %o5, 0x1f8, %g2

2759         srlx    %o5, 29, %g4
2760         ldx     [%l7 + %g2], %g2
2761         and     %g3, 0x1f8, %g3
```

```
2763         srlx    %o5, 12, %o2
2764         ldx     [%l1 + %g3], %g3
2765         and     %g4, 0x1f8, %g4

2767         srlx    %o5, 6, %o3
2768         ldx     [%l3 + %g4], %g4
2769         and     %o2, 0x1f8, %o2

2771         srlx    %o5, 35, %o4
2772         ldx     [%l5 + %o2], %o2
2773         and     %o3, 0x1f8, %o3

2775         or      %g1, %g2, %g2
2776         ldx     [%l6 + %o3], %o3
2777         and     %o4, 0x1f8, %o4

2779         srlx    %o5, 55, %o5
2780         ldx     [%l2 + %o4], %o4
2781         or      %g3, %g4, %g4

2783         or      %g2, %g4, %g4
2784         ldx     [%l0 + %o5], %o5
2785         or      %o2, %o3, %o3

2787         or      %g4, %o3, %o3
2788         ldx     [%i0], %g2
2789         subcc   %i4,1,%i4

2791         xor     %o0, %o3, %o0
2792         or      %o4, %o5, %o5

2794         xor     %o0, %g2, %g1
2795 !       bnz     %icc, .L2
2796         xor     %o0, %o5, %o0

2798         srlx    %g1, 55, %g1
2799         xor     %o0, %g2, %o5

2801         srlx    %o5, 41, %g3
2802         ldx     [%l0 + %g1], %g1
2803         and     %o5, 0x1f8, %g2

2805         srlx    %o5, 29, %g4
2806         ldx     [%l7 + %g2], %g2
2807         and     %g3, 0x1f8, %g3

2809         srlx    %o5, 12, %o2
2810         ldx     [%l1 + %g3], %g3
2811         and     %g4, 0x1f8, %g4

2813         srlx    %o5, 35, %o3
2814         ldx     [%l3 + %g4], %g4
2815         and     %o2, 0x1f8, %o2

2817         srlx    %o5, 6, %o4
2818         ldx     [%l5 + %o2], %o2
2819         and     %o3, 0x1f8, %o3

2821         or      %g1, %g2, %g2
2822         ldx     [%l2 + %o3], %o3
2823         and     %o4, 0x1f8, %o4

2825         srl     %o5, 23, %o5
2826         ldx     [%l6 + %o4], %o4
2827         or      %g3, %g4, %g4
```



```
2829         or      %g2, %g4, %g4
2830         ldx     [%l4 + %o5], %o5
2831         or      %o2, %o3, %o3

2833         or      %g4, %o3, %o3
2834         ldx     [%i0 + 8], %g2
2835         add     %i0, 16, %i0

2837         xor     %o1, %o3, %o1
2838         or      %o4, %o5, %o5

2840         xor     %o1, %g2, %g1
2841         xor     %o1, %o5, %o1

2843         srl     %g1, 23, %g1
2844         xor     %o1, %g2, %o5

2846         srlx    %o5, 41, %g3
2847         ldx     [%l4 + %g1], %g1
2848         and     %o5, 0x1f8, %g2

2850         srlx    %o5, 29, %g4
2851         ldx     [%l7 + %g2], %g2
2852         and     %g3, 0x1f8, %g3

2854         srlx    %o5, 12, %o2
2855         ldx     [%l1 + %g3], %g3
2856         and     %g4, 0x1f8, %g4

2858         srlx    %o5, 6, %o3
2859         ldx     [%l3 + %g4], %g4
2860         and     %o2, 0x1f8, %o2

2862         srlx    %o5, 35, %o4
2863         ldx     [%l5 + %o2], %o2
2864         and     %o3, 0x1f8, %o3

2866         or      %g1, %g2, %g2
2867         ldx     [%l6 + %o3], %o3
2868         and     %o4, 0x1f8, %o4

2870         srlx    %o5, 55, %o5
2871         ldx     [%l2 + %o4], %o4
2872         or      %g3, %g4, %g4

2874         or      %g2, %g4, %g4
2875         ldx     [%l0 + %o5], %o5
2876         or      %o2, %o3, %o3

2878         or      %g4, %o3, %o3
2879         ldx     [%i0], %g2
2880         subcc   %i4,1,%i4

2882         xor     %o0, %o3, %o0
2883         or      %o4, %o5, %o5

2885         xor     %o0, %g2, %g1
2886 !       bnz     %icc, .L2
2887         xor     %o0, %o5, %o0

2889         srlx    %g1, 55, %g1
2890         xor     %o0, %g2, %o5

2892         srlx    %o5, 41, %g3
2893         ldx     [%l0 + %g1], %g1
```

```
2894          and     %o5, 0x1f8, %g2

2896          srlx    %o5, 29, %g4
2897          ldx     [%l7 + %g2], %g2
2898          and     %g3, 0x1f8, %g3

2900          srlx    %o5, 12, %o2
2901          ldx     [%l1 + %g3], %g3
2902          and     %g4, 0x1f8, %g4

2904          srlx    %o5, 35, %o3
2905          ldx     [%l3 + %g4], %g4
2906          and     %o2, 0x1f8, %o2

2908          srlx    %o5, 6, %o4
2909          ldx     [%l5 + %o2], %o2
2910          and     %o3, 0x1f8, %o3

2912          or      %g1, %g2, %g2
2913          ldx     [%l2 + %o3], %o3
2914          and     %o4, 0x1f8, %o4

2916          srl     %o5, 23, %o5
2917          ldx     [%l6 + %o4], %o4
2918          or      %g3, %g4, %g4

2920          or      %g2, %g4, %g4
2921          ldx     [%l4 + %o5], %o5
2922          or      %o2, %o3, %o3

2924          or      %g4, %o3, %o3
2925          ldx     [%i0 + 8], %g2
2926          add     %i0, 16, %i0

2928          xor     %o1, %o3, %o1
2929          or      %o4, %o5, %o5

2931          xor     %o1, %g2, %g1
2932          xor     %o1, %o5, %o1

2934          srl     %g1, 23, %g1
2935          xor     %o1, %g2, %o5

2937          srlx    %o5, 41, %g3
2938          ldx     [%l4 + %g1], %g1
2939          and     %o5, 0x1f8, %g2

2941          srlx    %o5, 29, %g4
2942          ldx     [%l7 + %g2], %g2
2943          and     %g3, 0x1f8, %g3

2945          srlx    %o5, 12, %o2
2946          ldx     [%l1 + %g3], %g3
2947          and     %g4, 0x1f8, %g4

2949          srlx    %o5, 6, %o3
2950          ldx     [%l3 + %g4], %g4
2951          and     %o2, 0x1f8, %o2

2953          srlx    %o5, 35, %o4
2954          ldx     [%l5 + %o2], %o2
2955          and     %o3, 0x1f8, %o3

2957          or      %g1, %g2, %g2
2958          ldx     [%l6 + %o3], %o3
2959          and     %o4, 0x1f8, %o4
```



```
2961          srlx    %o5, 55, %o5
2962          ldx     [%l2 + %o4], %o4
2963          or      %g3, %g4, %g4

2965          or      %g2, %g4, %g4
2966          ldx     [%l0 + %o5], %o5
2967          or      %o2, %o3, %o3

2969          or      %g4, %o3, %o3
2970          ldx     [%i0], %g2
2971          subcc   %i4,1,%i4

2973          xor     %o0, %o3, %o0
2974          or      %o4, %o5, %o5

2976          xor     %o0, %g2, %g1
2977 !        bnz     %icc, .L2
2978          xor     %o0, %o5, %o0

2980          srlx    %g1, 55, %g1
2981          xor     %o0, %g2, %o5

2983          srlx    %o5, 41, %g3
2984          ldx     [%l0 + %g1], %g1
2985          and     %o5, 0x1f8, %g2

2987          srlx    %o5, 29, %g4
2988          ldx     [%l7 + %g2], %g2
2989          and     %g3, 0x1f8, %g3

2991          srlx    %o5, 12, %o2
2992          ldx     [%l1 + %g3], %g3
2993          and     %g4, 0x1f8, %g4

2995          srlx    %o5, 35, %o3
2996          ldx     [%l3 + %g4], %g4
2997          and     %o2, 0x1f8, %o2

2999          srlx    %o5, 6, %o4
3000          ldx     [%l5 + %o2], %o2
3001          and     %o3, 0x1f8, %o3

3003          or      %g1, %g2, %g2
3004          ldx     [%l2 + %o3], %o3
3005          and     %o4, 0x1f8, %o4

3007          srl     %o5, 23, %o5
3008          ldx     [%l6 + %o4], %o4
3009          or      %g3, %g4, %g4

3011          or      %g2, %g4, %g4
3012          ldx     [%l4 + %o5], %o5
3013          or      %o2, %o3, %o3

3015          or      %g4, %o3, %o3
3016          ldx     [%i0 + 8], %g2
3017          add     %i0, 16, %i0

3019          xor     %o1, %o3, %o1
3020          or      %o4, %o5, %o5

3022          xor     %o1, %g2, %g1
3023          xor     %o1, %o5, %o1

3025          srl     %g1, 23, %g1
```

```
3026         xor     %o1, %g2, %o5

3028         srlx    %o5, 41, %g3
3029         ldx     [%l4 + %g1], %g1
3030         and     %o5, 0x1f8, %g2

3032         srlx    %o5, 29, %g4
3033         ldx     [%l7 + %g2], %g2
3034         and     %g3, 0x1f8, %g3

3036         srlx    %o5, 12, %o2
3037         ldx     [%l1 + %g3], %g3
3038         and     %g4, 0x1f8, %g4

3040         srlx    %o5, 6, %o3
3041         ldx     [%l3 + %g4], %g4
3042         and     %o2, 0x1f8, %o2

3044         srlx    %o5, 35, %o4
3045         ldx     [%l5 + %o2], %o2
3046         and     %o3, 0x1f8, %o3

3048         or      %g1, %g2, %g2
3049         ldx     [%l6 + %o3], %o3
3050         and     %o4, 0x1f8, %o4

3052         srlx    %o5, 55, %o5
3053         ldx     [%l2 + %o4], %o4
3054         or      %g3, %g4, %g4

3056         or      %g2, %g4, %g4
3057         ldx     [%l0 + %o5], %o5
3058         or      %o2, %o3, %o3

3060         or      %g4, %o3, %o3
3061         ldx     [%i0], %g2
3062         subcc   %i4,1,%i4

3064         xor     %o0, %o3, %o0
3065         or      %o4, %o5, %o5

3067         xor     %o0, %g2, %g1
3068 !       bnz     %icc, .L2
3069         xor     %o0, %o5, %o0

3071         srlx    %g1, 55, %g1
3072         xor     %o0, %g2, %o5

3074         srlx    %o5, 41, %g3
3075         ldx     [%l0 + %g1], %g1
3076         and     %o5, 0x1f8, %g2

3078         srlx    %o5, 29, %g4
3079         ldx     [%l7 + %g2], %g2
3080         and     %g3, 0x1f8, %g3

3082         srlx    %o5, 12, %o2
3083         ldx     [%l1 + %g3], %g3
3084         and     %g4, 0x1f8, %g4

3086         srlx    %o5, 35, %o3
3087         ldx     [%l3 + %g4], %g4
3088         and     %o2, 0x1f8, %o2

3090         srlx    %o5, 6, %o4
3091         ldx     [%l5 + %o2], %o2
```



```
3092         and     %o3, 0x1f8, %o3

3094         or      %g1, %g2, %g2
3095         ldx     [%l2 + %o3], %o3
3096         and     %o4, 0x1f8, %o4

3098         srl     %o5, 23, %o5
3099         ldx     [%l6 + %o4], %o4
3100         or      %g3, %g4, %g4

3102         or      %g2, %g4, %g4
3103         ldx     [%l4 + %o5], %o5
3104         or      %o2, %o3, %o3

3106         or      %g4, %o3, %o3
3107         ldx     [%i0 + 8], %g2
3108         add     %i0, 16, %i0

3110         xor     %o1, %o3, %o1
3111         or      %o4, %o5, %o5

3113         xor     %o1, %g2, %g1
3114         xor     %o1, %o5, %o1

3116         srl     %g1, 23, %g1
3117         xor     %o1, %g2, %o5

3119         srlx    %o5, 41, %g3
3120         ldx     [%l4 + %g1], %g1
3121         and     %o5, 0x1f8, %g2

3123         srlx    %o5, 29, %g4
3124         ldx     [%l7 + %g2], %g2
3125         and     %g3, 0x1f8, %g3

3127         srlx    %o5, 12, %o2
3128         ldx     [%l1 + %g3], %g3
3129         and     %g4, 0x1f8, %g4

3131         srlx    %o5, 6, %o3
3132         ldx     [%l3 + %g4], %g4
3133         and     %o2, 0x1f8, %o2

3135         srlx    %o5, 35, %o4
3136         ldx     [%l5 + %o2], %o2
3137         and     %o3, 0x1f8, %o3

3139         or      %g1, %g2, %g2
3140         ldx     [%l6 + %o3], %o3
3141         and     %o4, 0x1f8, %o4

3143         srlx    %o5, 55, %o5
3144         ldx     [%l2 + %o4], %o4
3145         or      %g3, %g4, %g4

3147         or      %g2, %g4, %g4
3148         ldx     [%l0 + %o5], %o5
3149         or      %o2, %o3, %o3

3151         or      %g4, %o3, %o3
3152         ldx     [%i0], %g2
3153         subcc   %i4,1,%i4

3155         xor     %o0, %o3, %o0
3156         or      %o4, %o5, %o5
```

```
3158         xor     %o0, %g2, %g1
3159 !       bnz     %icc, .L2
3160         xor     %o0, %o5, %o0

3162         srlx    %g1, 55, %g1
3163         xor     %o0, %g2, %o5

3165         srlx    %o5, 41, %g3
3166         ldx     [%l0 + %g1], %g1
3167         and     %o5, 0x1f8, %g2

3169         srlx    %o5, 29, %g4
3170         ldx     [%l7 + %g2], %g2
3171         and     %g3, 0x1f8, %g3

3173         srlx    %o5, 12, %o2
3174         ldx     [%l1 + %g3], %g3
3175         and     %g4, 0x1f8, %g4

3177         srlx    %o5, 35, %o3
3178         ldx     [%l3 + %g4], %g4
3179         and     %o2, 0x1f8, %o2

3181         srlx    %o5, 6, %o4
3182         ldx     [%l5 + %o2], %o2
3183         and     %o3, 0x1f8, %o3

3185         or      %g1, %g2, %g2
3186         ldx     [%l2 + %o3], %o3
3187         and     %o4, 0x1f8, %o4

3189         srl     %o5, 23, %o5
3190         ldx     [%l6 + %o4], %o4
3191         or      %g3, %g4, %g4

3193         or      %g2, %g4, %g4
3194         ldx     [%l4 + %o5], %o5
3195         or      %o2, %o3, %o3

3197         or      %g4, %o3, %o3
3198         ldx     [%i0 + 8], %g2
3199 #ifdef __sparcv9
3200         subcc   %i2, 1, %i2             ! one_or_three for v9
3201 #else
3202         subcc   %i3, 1, %i3             ! one_or_three for v8
3203 #endif

3205         xor     %o1, %o3, %o1
3206         or      %o4, %o5, %o5

3208         xor     %o1, %g2, %g1
3209         xor     %o1, %o5, %g5

3211         srl     %g1, 23, %g1
3212         xor     %g5, %g2, %o5

3214         srlx    %o5, 41, %g3
3215         ldx     [%l4 + %g1], %g1
3216         and     %o5, 0x1f8, %g2

3218         srlx    %o5, 29, %g4
3219         ldx     [%l7 + %g2], %g2
3220         and     %g3, 0x1f8, %g3

3222         srlx    %o5, 12, %o2
3223         ldx     [%l1 + %g3], %g3
```



```
3224         and     %g4, 0x1f8, %g4

3226         srlx    %o5, 6, %o3
3227         ldx     [%l3 + %g4], %g4
3228         and     %o2, 0x1f8, %o2

3230         srlx    %o5, 35, %o4
3231         ldx     [%l5 + %o2], %o2
3232         and     %o3, 0x1f8, %o3

3234         or      %g1, %g2, %g2
3235         ldx     [%l6 + %o3], %o3
3236         and     %o4, 0x1f8, %o4

3238         srlx    %o5, 55, %o5
3239         ldx     [%l2 + %o4], %o4
3240         or      %g3, %g4, %g4

3242         or      %g2, %g4, %g4
3243         ldx     [%l0 + %o5], %o5
3244         bz,pn   %icc, .L4               ! if finished (one or three iterations
3245         or      %o2, %o3, %o3           ! of the 16 rounds), go to final perm.

3247         or      %g4, %o3, %o3
3248         ldx     [%i0 + 16], %g2
3249         add     %i0, 16, %i0

3251         xor     %o0, %o3, %o0
3252         or      %o4, %o5, %o4

3254         xor     %g5, %g2, %o5
3255         xor     %o0, %o4, %o1

3257         srlx    %o5, 55, %g1
3258         ba      .L3
3259         or      %g5, %g0, %o0

3261
3262 .L4:
3263         or      %g4, %o3, %o3
3264 #ifdef __sparcv9
3265         ldx     [%i5 + 16], %i5         ! &(dec_fp_table[0])
3266 #else
3267         ld      [%i5 + 8], %i5          ! &(dec_fp_table[0])
3268 #endif
3269         or      %g0, 0xf, %g4

3271         xor     %o0, %o3, %o0
3272         or      %o4, %o5, %o5

3274         sllx    %g4, 59, %g4            ! mask for bits 1-4
3275         xor     %o0, %o5, %o0

3277 ! fp starts here

3279         srlx    %o0, 33, %g1
3280         and     %o0, %g4, %g2

3282         srlx    %g5, 37, %l0
3283         and     %g5, %g4, %g3

3285         srlx    %g2, 53, %g2
3286         and     %g1, 0x3c0, %g1

3288         srlx    %g3, 57, %l1
3289         and     %l0, 0x3c, %l0
```

```
3291         srlx    %o0, 21, %g3
3292         or      %g1, %l0, %l0

3294         srlx    %g5, 25, %l2
3295         ld      [%i5 + %l0], %l0
3296         or      %l1, %g2, %l1

3298         srlx    %o0, 4, %g4
3299         ld      [%i5 + %l1], %l1
3300         and     %g3, 0x3c0, %g3

3302         srlx    %g5, 8, %l3
3303         and     %l2, 0x3c, %l2

3305         srlx    %o0, 27, %g1
3306         or      %l2, %g3, %l2

3308         srlx    %g5, 31, %l4
3309         ld      [%i5 + %l2], %l2
3310         and     %g4, 0x3c0, %g4

3312         and     %l3, 0x3c, %l3
3313         and     %g1, 0x3c0, %g1

3315         srlx    %o0, 10, %g2
3316         or      %l3, %g4, %l3

3318         srlx    %g5, 14, %l5
3319         ld      [%i5 + %l3], %l3
3320         and     %l4, 0x3c, %l4

3322         sllx    %o0, 2, %g3
3323         or      %l4, %g1, %l4

3325         srlx    %g5, 2, %l6
3326         ld      [%i5 + %l4], %l4
3327         and     %g2, 0x3c0, %g2

3329         srlx    %o0, 39, %g4
3330         and     %l5, 0x3c, %l5

3332         and     %g3, 0x3c0, %g3
3333         or      %l5, %g2, %l5

3335         srlx    %g5, 43, %l7
3336         ld      [%i5 + %l5], %l5
3337         and     %l6, 0x3c, %l6

3339         and     %g4, 0x3c0, %g4
3340         or      %l6, %g3, %l6

3342         srl     %l0, 2, %l0
3343         ld      [%i5 + %l6], %l6
3344         and     %l7, 0x3c, %l7

3346         srl     %l2, 4, %l2
3347         or      %l7, %g4, %l7

3349         srl     %l4, 2, %l4
3350         ld      [%i5 + %l7], %l7
3351         or      %l0, %l1, %l1

3353         srl     %l5, 4, %l5
3354         or      %l1, %l2, %l2
```



```
3356         srl     %l6, 6, %l6
3357         or      %l4, %l5, %l5

3359         srl     %l3, 6, %l3
3360         or      %l6, %l7, %l7

3362         or      %l2, %l3, %i1
3363         or      %l5, %l7, %i0

3365 !
3366 ! result at this point is in i0-i1, just as it should for v8
3367 !
3368 #ifdef __sparcv9
3369         srl     %i1, 0, %i1

3371         sllx    %i0, 32, %i0

3373         or      %i0, %i1, %i0
3374 #endif

3376
3377         ret
3378         restore %g0,%g0,%g0
3379         .type   des_crypt_impl,2
3380         .size   des_crypt_impl,(.-des_crypt_impl)

3382         .align  32
3383 !
3384 ! CONSTANT POOL
3385 !
3386 des_fp_table:
3387         .word   0
3388         .word   -2147483648
3389         .word   8388608
3390         .word   -2139095040
3391         .word   32768
3392         .word   -2147450880
3393         .word   8421376
3394         .word   -2139062272
3395         .word   128
3396         .word   -2147483520
3397         .word   8388736
3398         .word   -2139094912
3399         .word   32896
3400         .word   -2147450752
3401         .word   8421504
3402         .word   -2139062144
3403         .word   1073741824
3404         .word   -1073741824
3405         .word   1082130432
3406         .word   -1065353216
3407         .word   1073774592
3408         .word   -1073709056
3409         .word   1082163200
3410         .word   -1065320448
3411         .word   1073741952
3412         .word   -1073741696
3413         .word   1082130560
3414         .word   -1065353088
3415         .word   1073774720
3416         .word   -1073708928
3417         .word   1082163328
3418         .word   -1065320320
3419         .word   4194304
3420         .word   -2143289344
3421         .word   12582912
```

```
3422         .word   -2134900736
3423         .word   4227072
3424         .word   -2143256576
3425         .word   12615680
3426         .word   -2134867968
3427         .word   4194432
3428         .word   -2143289216
3429         .word   12583040
3430         .word   -2134900608
3431         .word   4227200
3432         .word   -2143256448
3433         .word   12615808
3434         .word   -2134867840
3435         .word   1077936128
3436         .word   -1069547520
3437         .word   1086324736
3438         .word   -1061158912
3439         .word   1077968896
3440         .word   -1069514752
3441         .word   1086357504
3442         .word   -1061126144
3443         .word   1077936256
3444         .word   -1069547392
3445         .word   1086324864
3446         .word   -1061158784
3447         .word   1077969024
3448         .word   -1069514624
3449         .word   1086357632
3450         .word   -1061126016
3451         .word   16384
3452         .word   -2147467264
3453         .word   8404992
3454         .word   -2139078656
3455         .word   49152
3456         .word   -2147434496
3457         .word   8437760
3458         .word   -2139045888
3459         .word   16512
3460         .word   -2147467136
3461         .word   8405120
3462         .word   -2139078528
3463         .word   49280
3464         .word   -2147434368
3465         .word   8437888
3466         .word   -2139045760
3467         .word   1073758208
3468         .word   -1073725440
3469         .word   1082146816
3470         .word   -1065336832
3471         .word   1073790976
3472         .word   -1073692672
3473         .word   1082179584
3474         .word   -1065304064
3475         .word   1073758336
3476         .word   -1073725312
3477         .word   1082146944
3478         .word   -1065336704
3479         .word   1073791104
3480         .word   -1073692544
3481         .word   1082179712
3482         .word   -1065303936
3483         .word   4210688
3484         .word   -2143272960
3485         .word   12599296
3486         .word   -2134884352
3487         .word   4243456
```



```
3488         .word   -2143240192
3489         .word   12632064
3490         .word   -2134851584
3491         .word   4210816
3492         .word   -2143272832
3493         .word   12599424
3494         .word   -2134884224
3495         .word   4243584
3496         .word   -2143240064
3497         .word   12632192
3498         .word   -2134851456
3499         .word   1077952512
3500         .word   -1069531136
3501         .word   1086341120
3502         .word   -1061142528
3503         .word   1077985280
3504         .word   -1069498368
3505         .word   1086373888
3506         .word   -1061109760
3507         .word   1077952640
3508         .word   -1069531008
3509         .word   1086341248
3510         .word   -1061142400
3511         .word   1077985408
3512         .word   -1069498240
3513         .word   1086374016
3514         .word   -1061109632
3515         .word   64
3516         .word   -2147483584
3517         .word   8388672
3518         .word   -2139094976
3519         .word   32832
3520         .word   -2147450816
3521         .word   8421440
3522         .word   -2139062208
3523         .word   192
3524         .word   -2147483456
3525         .word   8388800
3526         .word   -2139094848
3527         .word   32960
3528         .word   -2147450688
3529         .word   8421568
3530         .word   -2139062080
3531         .word   1073741888
3532         .word   -1073741760
3533         .word   1082130496
3534         .word   -1065353152
3535         .word   1073774656
3536         .word   -1073708992
3537         .word   1082163264
3538         .word   -1065320384
3539         .word   1073742016
3540         .word   -1073741632
3541         .word   1082130624
3542         .word   -1065353024
3543         .word   1073774784
3544         .word   -1073708864
3545         .word   1082163392
3546         .word   -1065320256
3547         .word   4194368
3548         .word   -2143289280
3549         .word   12582976
3550         .word   -2134900672
3551         .word   4227136
3552         .word   -2143256512
3553         .word   12615744
```

```
3554         .word   -2134867904
3555         .word   4194496
3556         .word   -2143289152
3557         .word   12583104
3558         .word   -2134900544
3559         .word   4227264
3560         .word   -2143256384
3561         .word   12615872
3562         .word   -2134867776
3563         .word   1077936192
3564         .word   -1069547456
3565         .word   1086324800
3566         .word   -1061158848
3567         .word   1077968960
3568         .word   -1069514688
3569         .word   1086357568
3570         .word   -1061126080
3571         .word   1077936320
3572         .word   -1069547328
3573         .word   1086324928
3574         .word   -1061158720
3575         .word   1077969088
3576         .word   -1069514560
3577         .word   1086357696
3578         .word   -1061125952
3579         .word   16448
3580         .word   -2147467200
3581         .word   8405056
3582         .word   -2139078592
3583         .word   49216
3584         .word   -2147434432
3585         .word   8437824
3586         .word   -2139045824
3587         .word   16576
3588         .word   -2147467072
3589         .word   8405184
3590         .word   -2139078464
3591         .word   49344
3592         .word   -2147434304
3593         .word   8437952
3594         .word   -2139045696
3595         .word   1073758272
3596         .word   -1073725376
3597         .word   1082146880
3598         .word   -1065336768
3599         .word   1073791040
3600         .word   -1073692608
3601         .word   1082179648
3602         .word   -1065304000
3603         .word   1073758400
3604         .word   -1073725248
3605         .word   1082147008
3606         .word   -1065336640
3607         .word   1073791168
3608         .word   -1073692480
3609         .word   1082179776
3610         .word   -1065303872
3611         .word   4210752
3612         .word   -2143272896
3613         .word   12599360
3614         .word   -2134884288
3615         .word   4243520
3616         .word   -2143240128
3617         .word   12632128
3618         .word   -2134851520
3619         .word   4210880
```



```
3620         .word   -2143272768
3621         .word   12599488
3622         .word   -2134884160
3623         .word   4243648
3624         .word   -2143240000
3625         .word   12632256
3626         .word   -2134851392
3627         .word   1077952576
3628         .word   -1069531072
3629         .word   1086341184
3630         .word   -1061142464
3631         .word   1077985344
3632         .word   -1069498304
3633         .word   1086373952
3634         .word   -1061109696
3635         .word   1077952704
3636         .word   -1069530944
3637         .word   1086341312
3638         .word   -1061142336
3639         .word   1077985472
3640         .word   -1069498176
3641         .word   1086374080
3642         .word   -1061109568
3643         .type   des_fp_table,#object
3644         .size   des_fp_table,1024

3652 /* EXPORT DELETE END */

3646 #endif  /* lint || __lint */
```

```
**********************************************************
    1061 Thu Jul 11 01:29:08 2013
new/usr/src/common/crypto/rsa/Makefile
first pass
**********************************************************
```

```
#
# 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 2003 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# ident "%Z%%M% %I%     %E% SMI"
#
# common/crypto/rsa/Makefile
#
# include global definitions
include $(SRC)/Makefile.master

.KEEP_STATE:

FRC:

# EXPORT DELETE START
EXPORT_SRC:
        $(RM) Makefile+ rsa_impl.c+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < rsa_impl.c > rsa_impl.c+
        $(MV) rsa_impl.c+ rsa_impl.c
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(RM) Makefile
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile rsa_impl.c
# EXPORT DELETE END
```

```
**********************************************************
   16725 Thu Jul 11 01:29:09 2013
new/usr/src/common/crypto/rsa/rsa_impl.c
first pass
**********************************************************
______unchanged_portion_omitted_

 106 /* psize and qsize are in bits */
 107 static BIG_ERR_CODE
 108 RSA_key_init(RSAkey *key, int psize, int qsize)
 109 {
 110         BIG_ERR_CODE err = BIG_OK;

 112 /* EXPORT DELETE START */

 112         int plen, qlen, nlen;

 114         plen = BITLEN2BIGNUMLEN(psize);
 115         qlen = BITLEN2BIGNUMLEN(qsize);
 116         nlen = plen + qlen;
 117         key->size = psize + qsize;
 118         if ((err = big_init(&(key->p), plen)) != BIG_OK)
 119                 return (err);
 120         if ((err = big_init(&(key->q), qlen)) != BIG_OK)
 121                 goto ret1;
 122         if ((err = big_init(&(key->n), nlen)) != BIG_OK)
 123                 goto ret2;
 124         if ((err = big_init(&(key->d), nlen)) != BIG_OK)
 125                 goto ret3;
 126         if ((err = big_init(&(key->e), nlen)) != BIG_OK)
 127                 goto ret4;
 128         if ((err = big_init(&(key->dmodpminus1), plen)) != BIG_OK)
 129                 goto ret5;
 130         if ((err = big_init(&(key->dmodqminus1), qlen)) != BIG_OK)
 131                 goto ret6;
 132         if ((err = big_init(&(key->pinvmodq), qlen)) != BIG_OK)
 133                 goto ret7;
 134         if ((err = big_init(&(key->p_rr), plen)) != BIG_OK)
 135                 goto ret8;
 136         if ((err = big_init(&(key->q_rr), qlen)) != BIG_OK)
 137                 goto ret9;
 138         if ((err = big_init(&(key->n_rr), nlen)) != BIG_OK)
 139                 goto ret10;

 141         return (BIG_OK);

 143 ret10:
 144         big_finish(&(key->q_rr));
 145 ret9:
 146         big_finish(&(key->p_rr));
 147 ret8:
 148         big_finish(&(key->pinvmodq));
 149 ret7:
 150         big_finish(&(key->dmodqminus1));
 151 ret6:
 152         big_finish(&(key->dmodpminus1));
 153 ret5:
 154         big_finish(&(key->e));
 155 ret4:
 156         big_finish(&(key->d));
 157 ret3:
 158         big_finish(&(key->n));
 159 ret2:
 160         big_finish(&(key->q));
 161 ret1:
 162         big_finish(&(key->p));
```



```
 166 /* EXPORT DELETE END */

 164         return (err);
 165 }

 167 static void
 168 RSA_key_finish(RSAkey *key)
 169 {

 175 /* EXPORT DELETE START */

 170         big_finish(&(key->n_rr));
 171         big_finish(&(key->q_rr));
 172         big_finish(&(key->p_rr));
 173         big_finish(&(key->pinvmodq));
 174         big_finish(&(key->dmodqminus1));
 175         big_finish(&(key->dmodpminus1));
 176         big_finish(&(key->e));
 177         big_finish(&(key->d));
 178         big_finish(&(key->n));
 179         big_finish(&(key->q));
 180         big_finish(&(key->p));

 189 /* EXPORT DELETE END */

 181 }

 183 /*
 184  * Generate RSA key
 185  */
 186 static CK_RV
 187 generate_rsa_key(RSAkey *key, int psize, int qsize, BIGNUM *pubexp,
 188     int (*rfunc)(void *, size_t))
 189 {
 190         CK_RV           rv = CKR_OK;

 202 /* EXPORT DELETE START */

 192         int             (*rf)(void *, size_t);
 193         BIGNUM          a, b, c, d, e, f, g, h;
 194         int             len, keylen, size;
 195         BIG_ERR_CODE    brv = BIG_OK;

 197         size = psize + qsize;
 198         keylen = BITLEN2BIGNUMLEN(size);
 199         len = keylen * 2 + 1;
 200         key->size = size;

 202         /*
 203          * Note: It is not really necessary to compute e, it is in pubexp:
 204          *      (void) big_copy(&(key->e), pubexp);
 205          */

 207         a.malloced = 0;
 208         b.malloced = 0;
 209         c.malloced = 0;
 210         d.malloced = 0;
 211         e.malloced = 0;
 212         f.malloced = 0;
 213         g.malloced = 0;
 214         h.malloced = 0;

 216         if ((big_init(&a, len) != BIG_OK) ||
 217             (big_init(&b, len) != BIG_OK) ||
 218             (big_init(&c, len) != BIG_OK) ||
```

```
219             (big_init(&d, len) != BIG_OK) ||
220             (big_init(&e, len) != BIG_OK) ||
221             (big_init(&f, len) != BIG_OK) ||
222             (big_init(&g, len) != BIG_OK) ||
223             (big_init(&h, len) != BIG_OK)) {
224                 big_finish(&h);
225                 big_finish(&g);
226                 big_finish(&f);
227                 big_finish(&e);
228                 big_finish(&d);
229                 big_finish(&c);
230                 big_finish(&b);
231                 big_finish(&a);

233                 return (CKR_HOST_MEMORY);
234         }

236         rf = rfunc;
237         if (rf == NULL) {
238 #ifdef _KERNEL
239                 rf = (int (*)(void *, size_t))random_get_pseudo_bytes;
240 #else
241                 rf = pkcs11_get_urandom;
242 #endif
243         }

245 nextp:
246         if ((brv = big_random(&a, psize, rf)) != BIG_OK) {
247                 goto ret;
248         }

250         if ((brv = big_nextprime_pos(&b, &a)) != BIG_OK) {
251                 goto ret;
252         }
253         /* b now contains the potential prime p */

255         (void) big_sub_pos(&a, &b, &big_One);
256         if ((brv = big_ext_gcd_pos(&f, &d, &g, pubexp, &a)) != BIG_OK) {
257                 goto ret;
258         }
259         if (big_cmp_abs(&f, &big_One) != 0) {
260                 goto nextp;
261         }

263         if ((brv = big_random(&c, qsize, rf)) != BIG_OK) {
264                 goto ret;
265         }

267 nextq:
268         (void) big_add(&a, &c, &big_Two);

270         if (big_bitlength(&a) != qsize) {
271                 goto nextp;
272         }
273         if (big_cmp_abs(&a, &b) == 0) {
274                 goto nextp;
275         }
276         if ((brv = big_nextprime_pos(&c, &a)) != BIG_OK) {
277                 goto ret;
278         }
279         /* c now contains the potential prime q */

281         if ((brv = big_mul(&g, &b, &c)) != BIG_OK) {
282                 goto ret;
283         }
284         if (big_bitlength(&g) != size) {
```



```
285                 goto nextp;
286         }
287         /* g now contains the potential modulus n */

289         (void) big_sub_pos(&a, &b, &big_One);
290         (void) big_sub_pos(&d, &c, &big_One);

292         if ((brv = big_mul(&a, &a, &d)) != BIG_OK) {
293                 goto ret;
294         }
295         if ((brv = big_ext_gcd_pos(&f, &d, &h, pubexp, &a)) != BIG_OK) {
296                 goto ret;
297         }
298         if (big_cmp_abs(&f, &big_One) != 0) {
299                 goto nextq;
300         } else {
301                 (void) big_copy(&e, pubexp);
302         }
303         if (d.sign == -1) {
304                 if ((brv = big_add(&d, &d, &a)) != BIG_OK) {
305                         goto ret;
306                 }
307         }
308         (void) big_copy(&(key->p), &b);
309         (void) big_copy(&(key->q), &c);
310         (void) big_copy(&(key->n), &g);
311         (void) big_copy(&(key->d), &d);
312         (void) big_copy(&(key->e), &e);

314         if ((brv = big_ext_gcd_pos(&a, &f, &h, &b, &c)) != BIG_OK) {
315                 goto ret;
316         }
317         if (f.sign == -1) {
318                 if ((brv = big_add(&f, &f, &c)) != BIG_OK) {
319                         goto ret;
320                 }
321         }
322         (void) big_copy(&(key->pinvmodq), &f);

324         (void) big_sub(&a, &b, &big_One);
325         if ((brv = big_div_pos(&a, &f, &d, &a)) != BIG_OK) {
326                 goto ret;
327         }
328         (void) big_copy(&(key->dmodpminus1), &f);
329         (void) big_sub(&a, &c, &big_One);
330         if ((brv = big_div_pos(&a, &f, &d, &a)) != BIG_OK) {
331                 goto ret;
332         }
333         (void) big_copy(&(key->dmodqminus1), &f);

335         /* pairwise consistency check:  decrypt and encrypt restores value */
336         if ((brv = big_random(&h, size, rf)) != BIG_OK) {
337                 goto ret;
338         }
339         if ((brv = big_div_pos(&a, &h, &h, &g)) != BIG_OK) {
340                 goto ret;
341         }
342         if ((brv = big_modexp(&a, &h, &d, &g, NULL)) != BIG_OK) {
343                 goto ret;
344         }

346         if ((brv = big_modexp(&b, &a, &e, &g, NULL)) != BIG_OK) {
347                 goto ret;
348         }

350         if (big_cmp_abs(&b, &h) != 0) {
```

```
351                 /* this should not happen */
352                 rv = generate_rsa_key(key, psize, qsize, pubexp, rf);
353                 goto ret1;
354         } else {
355                 brv = BIG_OK;
356         }

358 ret:
359         rv = convert_rv(brv);
360 ret1:
361         big_finish(&h);
362         big_finish(&g);
363         big_finish(&f);
364         big_finish(&e);
365         big_finish(&d);
366         big_finish(&c);
367         big_finish(&b);
368         big_finish(&a);

382 /* EXPORT DELETE END */

370         return (rv);
371 }

373 CK_RV
374 rsa_genkey_pair(RSAbytekey *bkey)
375 {
376         /*
377          * NOTE:  Whomever originally wrote this function swapped p and q.
378          * This table shows the mapping between name convention used here
379          * versus what is used in most texts that describe RSA key generation.
380          *      This function:                  Standard convention:
381          *      --------------                  --------------------
382          *      modulus, n                      -same-
383          *      prime 1, q                      prime 1, p
384          *      prime 2, p                      prime 2, q
385          *      private exponent, d             -same-
386          *      public exponent, e              -same-
387          *      exponent 1, d mod (q-1)         d mod (p-1)
388          *      exponent 2, d mod (p-1)         d mod (q-1)
389          *      coefficient, p^-1 mod q         q^-1 mod p
390          *
391          * Also notice the struct member for coefficient is named .pinvmodq
392          * rather than .qinvmodp, reflecting the switch.
393          *
394          * The code here wasn't unswapped, because "it works".  Further,
395          * p and q are interchangeable as long as exponent 1 and 2 and
396          * the coefficient are kept straight too.  This note is here to
397          * make the reader aware of the switcheroo.
398          */
399         CK_RV   rv = CKR_OK;

415 /* EXPORT DELETE START */

401         BIGNUM  public_exponent = {0};
402         RSAkey  rsakey;
403         uint32_t modulus_bytes;

405         if (bkey == NULL)
406                 return (CKR_ARGUMENTS_BAD);

408         /* Must have modulus bits set */
409         if (bkey->modulus_bits == 0)
410                 return (CKR_ARGUMENTS_BAD);

412         /* Must have public exponent set */
```



```
413         if (bkey->pubexpo_bytes == 0 || bkey->pubexpo == NULL)
414                 return (CKR_ARGUMENTS_BAD);

416         /* Note: modulus_bits may not be same as (8 * sizeof (modulus)) */
417         modulus_bytes = CRYPTO_BITS2BYTES(bkey->modulus_bits);

419         /* Modulus length needs to be between min key size and max key size. */
420         if ((modulus_bytes < MIN_RSA_KEYLENGTH_IN_BYTES) ||
421             (modulus_bytes > MAX_RSA_KEYLENGTH_IN_BYTES)) {
422                 return (CKR_KEY_SIZE_RANGE);
423         }

425         /*
426          * Initialize the RSA key.
427          */
428         if (RSA_key_init(&rsakey, modulus_bytes * 4, modulus_bytes * 4) !=
429             BIG_OK) {
430                 return (CKR_HOST_MEMORY);
431         }

433         /* Create a public exponent in bignum format. */
434         if (big_init(&public_exponent,
435             CHARLEN2BIGNUMLEN(bkey->pubexpo_bytes)) != BIG_OK) {
436                 rv = CKR_HOST_MEMORY;
437                 goto clean1;
438         }
439         bytestring2bignum(&public_exponent, bkey->pubexpo, bkey->pubexpo_bytes);

441         /* Generate RSA key pair. */
442         if ((rv = generate_rsa_key(&rsakey,
443             modulus_bytes * 4, modulus_bytes * 4, &public_exponent,
444             bkey->rfunc)) != CKR_OK) {
445                 big_finish(&public_exponent);
446                 goto clean1;
447         }
448         big_finish(&public_exponent);

450         /* modulus_bytes = rsakey.n.len * (int)sizeof (BIG_CHUNK_TYPE); */
451         bignum2bytestring(bkey->modulus, &(rsakey.n), modulus_bytes);

453         bkey->privexpo_bytes = rsakey.d.len * (int)sizeof (BIG_CHUNK_TYPE);
454         bignum2bytestring(bkey->privexpo, &(rsakey.d), bkey->privexpo_bytes);

456         bkey->pubexpo_bytes = rsakey.e.len * (int)sizeof (BIG_CHUNK_TYPE);
457         bignum2bytestring(bkey->pubexpo, &(rsakey.e), bkey->pubexpo_bytes);

459         bkey->prime1_bytes = rsakey.q.len * (int)sizeof (BIG_CHUNK_TYPE);
460         bignum2bytestring(bkey->prime1, &(rsakey.q), bkey->prime1_bytes);

462         bkey->prime2_bytes = rsakey.p.len * (int)sizeof (BIG_CHUNK_TYPE);
463         bignum2bytestring(bkey->prime2, &(rsakey.p), bkey->prime2_bytes);

465         bkey->expo1_bytes =
466             rsakey.dmodqminus1.len * (int)sizeof (BIG_CHUNK_TYPE);
467         bignum2bytestring(bkey->expo1, &(rsakey.dmodqminus1),
468             bkey->expo1_bytes);

470         bkey->expo2_bytes =
471             rsakey.dmodpminus1.len * (int)sizeof (BIG_CHUNK_TYPE);
472         bignum2bytestring(bkey->expo2,
473             &(rsakey.dmodpminus1), bkey->expo2_bytes);

475         bkey->coeff_bytes =
476             rsakey.pinvmodq.len * (int)sizeof (BIG_CHUNK_TYPE);
477         bignum2bytestring(bkey->coeff, &(rsakey.pinvmodq), bkey->coeff_bytes);
```

```
clean1:
        RSA_key_finish(&rsakey);

/* EXPORT DELETE END */

        return (rv);
}

/*
 * RSA encrypt operation
 */
CK_RV
rsa_encrypt(RSAbytekey *bkey, uchar_t *in, uint32_t in_len, uchar_t *out)
{
        CK_RV rv = CKR_OK;

/* EXPORT DELETE START */

        BIGNUM msg;
        RSAkey rsakey;
        uint32_t modulus_bytes;

        if (bkey == NULL)
                return (CKR_ARGUMENTS_BAD);

        /* Must have modulus and public exponent set */
        if (bkey->modulus_bits == 0 || bkey->modulus == NULL ||
            bkey->pubexpo_bytes == 0 || bkey->pubexpo == NULL)
                return (CKR_ARGUMENTS_BAD);

        /* Note: modulus_bits may not be same as (8 * sizeof (modulus)) */
        modulus_bytes = CRYPTO_BITS2BYTES(bkey->modulus_bits);

        if (bkey->pubexpo_bytes > modulus_bytes) {
                return (CKR_KEY_SIZE_RANGE);
        }

        /* psize and qsize for RSA_key_init is in bits. */
        if (RSA_key_init(&rsakey, modulus_bytes * 4, modulus_bytes * 4) !=
            BIG_OK) {
                return (CKR_HOST_MEMORY);
        }

        /* Size for big_init is in BIG_CHUNK_TYPE words. */
        if (big_init(&msg, CHARLEN2BIGNUMLEN(in_len)) != BIG_OK) {
                rv = CKR_HOST_MEMORY;
                goto clean2;
        }
        bytestring2bignum(&msg, in, in_len);

        /* Convert public exponent and modulus to big integer format. */
        bytestring2bignum(&(rsakey.e), bkey->pubexpo, bkey->pubexpo_bytes);
        bytestring2bignum(&(rsakey.n), bkey->modulus, modulus_bytes);

        if (big_cmp_abs(&msg, &(rsakey.n)) > 0) {
                rv = CKR_DATA_LEN_RANGE;
                goto clean3;
        }

        /* Perform RSA computation on big integer input data. */
        if (big_modexp(&msg, &msg, &(rsakey.e), &(rsakey.n), NULL) !=
            BIG_OK) {
                rv = CKR_HOST_MEMORY;
                goto clean3;
        }
```



```
        /* Convert the big integer output data to octet string. */
        bignum2bytestring(out, &msg, modulus_bytes);

clean3:
        big_finish(&msg);
clean2:
        RSA_key_finish(&rsakey);

/* EXPORT DELETE END */

        return (rv);
}

/*
 * RSA decrypt operation
 */
CK_RV
rsa_decrypt(RSAbytekey *bkey, uchar_t *in, uint32_t in_len, uchar_t *out)
{
        CK_RV rv = CKR_OK;

/* EXPORT DELETE START */

        BIGNUM msg;
        RSAkey rsakey;
        uint32_t modulus_bytes;

        if (bkey == NULL)
                return (CKR_ARGUMENTS_BAD);

        /* Must have modulus, prime1, prime2, expo1, expo2, and coeff set */
        if (bkey->modulus_bits == 0 || bkey->modulus == NULL ||
            bkey->prime1_bytes == 0 || bkey->prime1 == NULL ||
            bkey->prime2_bytes == 0 || bkey->prime2 == NULL ||
            bkey->expo1_bytes == 0 || bkey->expo1 == NULL ||
            bkey->expo2_bytes == 0 || bkey->expo2 == NULL ||
            bkey->coeff_bytes == 0 || bkey->coeff == NULL)
                return (CKR_ARGUMENTS_BAD);

        /* Note: modulus_bits may not be same as (8 * sizeof (modulus)) */
        modulus_bytes = CRYPTO_BITS2BYTES(bkey->modulus_bits);

        /* psize and qsize for RSA_key_init is in bits. */
        if (RSA_key_init(&rsakey, CRYPTO_BYTES2BITS(bkey->prime2_bytes),
            CRYPTO_BYTES2BITS(bkey->prime1_bytes)) != BIG_OK) {
                return (CKR_HOST_MEMORY);
        }

        /* Size for big_init is in BIG_CHUNK_TYPE words. */
        if (big_init(&msg, CHARLEN2BIGNUMLEN(in_len)) != BIG_OK) {
                rv = CKR_HOST_MEMORY;
                goto clean3;
        }
        /* Convert octet string input data to big integer format. */
        bytestring2bignum(&msg, in, in_len);

        /* Convert octet string modulus to big integer format. */
        bytestring2bignum(&(rsakey.n), bkey->modulus, modulus_bytes);

        if (big_cmp_abs(&msg, &(rsakey.n)) > 0) {
                rv = CKR_DATA_LEN_RANGE;
                goto clean4;
        }

        /* Convert the rest of private key attributes to big integer format. */
        bytestring2bignum(&(rsakey.q), bkey->prime1, bkey->prime1_bytes);
```

```
603         bytestring2bignum(&(rsakey.p), bkey->prime2, bkey->prime2_bytes);
604         bytestring2bignum(&(rsakey.dmodqminus1),
605             bkey->expo1, bkey->expo1_bytes);
606         bytestring2bignum(&(rsakey.dmodpminus1),
607             bkey->expo2, bkey->expo2_bytes);
608         bytestring2bignum(&(rsakey.pinvmodq),
609             bkey->coeff, bkey->coeff_bytes);

611         if ((big_cmp_abs(&(rsakey.dmodpminus1), &(rsakey.p)) > 0) ||
612             (big_cmp_abs(&(rsakey.dmodqminus1), &(rsakey.q)) > 0) ||
613             (big_cmp_abs(&(rsakey.pinvmodq), &(rsakey.q)) > 0)) {
614                 rv = CKR_KEY_SIZE_RANGE;
615                 goto clean4;
616         }

618         /* Perform RSA computation on big integer input data. */
619         if (big_modexp_crt(&msg, &msg, &(rsakey.dmodpminus1),
620             &(rsakey.dmodqminus1), &(rsakey.p), &(rsakey.q),
621             &(rsakey.pinvmodq), NULL, NULL) != BIG_OK) {
622                 rv = CKR_HOST_MEMORY;
623                 goto clean4;
624         }

626         /* Convert the big integer output data to octet string. */
627         bignum2bytestring(out, &msg, modulus_bytes);

629 clean4:
630         big_finish(&msg);
631 clean3:
632         RSA_key_finish(&rsakey);

658 /* EXPORT DELETE END */

634         return (rv);
635 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   54497 Thu Jul 11 01:29:10 2013
new/usr/src/common/net/wanboot/crypt/aes.c
first pass
**********************************************************
```

```
   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 2002-2003 Sun Microsystems, Inc.  All rights reserved.
  24  * Use is subject to license terms.
  25  */

  27 #pragma ident   "%Z%%M% %I%     %E% SMI"

  29 /*
  30  * AES implementation taken from public domain. The S-boxes
  31  * used by this implmentation are defined by NIST.
  32  *
  33  * For more information on AES refer to
  34  * http://csrc.nist.gov/CryptoToolkit/aes
  35  */

  37 #include <stdlib.h>
  38 #include <sys/sysmacros.h>

  40 #include "aes.h"

  42 /* Yay for Big-Endian Algorithms! */
  43 #ifdef _LITTLE_ENDIAN
  44 #define BSWAP_L(l) (((l & 0xff) << 24) | ((l & 0xff00) <<8) \
  45         | ((l & 0xff0000) >> 8) | ((l & 0xff000000) >>24))
  46 #else
  47 #define BSWAP_L(l) (l)
  48 #endif

  50 #define GETU32(p) BSWAP_L(*(uint32_t *)(p))
  51 #define PUTU32(ct, st)  *((uint32_t *)(ct)) = BSWAP_L(st)


  54 /* EXPORT DELETE START */
  54 /*
  55  * Te0[x] = S [x].[02, 01, 01, 03];
  56  * Te1[x] = S [x].[03, 02, 01, 01];
  57  * Te2[x] = S [x].[01, 03, 02, 01];
  58  * Te3[x] = S [x].[01, 01, 03, 02];
  59  * Te4[x] = S [x].[01, 01, 01, 01];
  60  *
```



```
  61  * Td0[x] = Si[x].[0e, 09, 0d, 0b];
  62  * Td1[x] = Si[x].[0b, 0e, 09, 0d];
  63  * Td2[x] = Si[x].[0d, 0b, 0e, 09];
  64  * Td3[x] = Si[x].[09, 0d, 0b, 0e];
  65  * Td4[x] = Si[x].[01, 01, 01, 01];
  66  */


  69 /* S-boxes */
  70 static const uint32_t Te0[256] = {
  71     0xc66363a5U, 0xf87c7c84U, 0xee777799U, 0xf67b7b8dU,
  72     0xfff2f20dU, 0xd66b6bbdU, 0xde6f6fb1U, 0x91c5c554U,
  73     0x60303050U, 0x02010103U, 0xce6767a9U, 0x562b2b7dU,
  74     0xe7fefe19U, 0xb5d7d762U, 0x4dababe6U, 0xec76769aU,
  75     0x8fcaca45U, 0x1f82829dU, 0x89c9c940U, 0xfa7d7d87U,
  76     0xeffafa15U, 0xb25959ebU, 0x8e4747c9U, 0xfbf0f00bU,
  77     0x41adadecU, 0xb3d4d467U, 0x5fa2a2fdU, 0x45afafeaU,
  78     0x239c9cbfU, 0x53a4a4f7U, 0xe4727296U, 0x9bc0c05bU,
  79     0x75b7b7c2U, 0xe1fdfd1cU, 0x3d9393aeU, 0x4c26266aU,
  80     0x6c36365aU, 0x7e3f3f41U, 0xf5f7f702U, 0x83cccc4fU,
  81     0x6834345cU, 0x51a5a5f4U, 0xd1e5e534U, 0xf9f1f108U,
  82     0xe2717193U, 0xabd8d873U, 0x62313153U, 0x2a15153fU,
  83     0x0804040cU, 0x95c7c752U, 0x46232365U, 0x9dc3c35eU,
  84     0x30181828U, 0x379696a1U, 0x0a05050fU, 0x2f9a9ab5U,
  85     0x0e070709U, 0x24121236U, 0x1b80809bU, 0xdfe2e23dU,
  86     0xcdebeb26U, 0x4e272769U, 0x7fb2b2cdU, 0xea75759fU,
  87     0x1209091bU, 0x1d83839eU, 0x582c2c74U, 0x341a1a2eU,
  88     0x361b1b2dU, 0xdc6e6eb2U, 0xb45a5aeeU, 0x5ba0a0fbU,
  89     0xa45252f6U, 0x763b3b4dU, 0xb7d6d661U, 0x7db3b3ceU,
  90     0x5229297bU, 0xdde3e33eU, 0x5e2f2f71U, 0x13848497U,
  91     0xa65353f5U, 0xb9d1d168U, 0x00000000U, 0xc1eded2cU,
  92     0x40202060U, 0xe3fcfc1fU, 0x79b1b1c8U, 0xb65b5bedU,
  93     0xd46a6abeU, 0x8dcbcb46U, 0x67bebed9U, 0x7239394bU,
  94     0x944a4adeU, 0x984c4cd4U, 0xb05858e8U, 0x85cfcf4aU,
  95     0xbbd0d06bU, 0xc5efef2aU, 0x4faaaae5U, 0xedfbfb16U,
  96     0x864343c5U, 0x9a4d4dd7U, 0x66333355U, 0x11858594U,
  97     0x8a4545cfU, 0xe9f9f910U, 0x04020206U, 0xfe7f7f81U,
  98     0xa05050f0U, 0x783c3c44U, 0x259f9fbaU, 0x4ba8a8e3U,
  99     0xa25151f3U, 0x5da3a3feU, 0x804040c0U, 0x058f8f8aU,
 100     0x3f9292adU, 0x219d9dbcU, 0x70383848U, 0xf1f5f504U,
 101     0x63bcbcdfU, 0x77b6b6c1U, 0xafdada75U, 0x42212163U,
 102     0x20101030U, 0xe5ffff1aU, 0xfdf3f30eU, 0xbfd2d26dU,
 103     0x81cdcd4cU, 0x180c0c14U, 0x26131335U, 0xc3ecec2fU,
 104     0xbe5f5fe1U, 0x359797a2U, 0x884444ccU, 0x2e171739U,
 105     0x93c4c457U, 0x55a7a7f2U, 0xfc7e7e82U, 0x7a3d3d47U,
 106     0xc86464acU, 0xba5d5de7U, 0x3219192bU, 0xe6737395U,
 107     0xc06060a0U, 0x19818198U, 0x9e4f4fd1U, 0xa3dcdc7fU,
 108     0x44222266U, 0x542a2a7eU, 0x3b9090abU, 0x0b888883U,
 109     0x8c4646caU, 0xc7eeee29U, 0x6bb8b8d3U, 0x2814143cU,
 110     0xa7dede79U, 0xbc5e5ee2U, 0x160b0b1dU, 0xaddbdb76U,
 111     0xdbe0e03bU, 0x64323256U, 0x743a3a4eU, 0x140a0a1eU,
 112     0x924949dbU, 0x0c06060aU, 0x4824246cU, 0xb85c5ce4U,
 113     0x9fc2c25dU, 0xbdd3d36eU, 0x43acacefU, 0xc46262a6U,
 114     0x399191a8U, 0x319595a4U, 0xd3e4e437U, 0xf279798bU,
 115     0xd5e7e732U, 0x8bc8c843U, 0x6e373759U, 0xda6d6db7U,
 116     0x018d8d8cU, 0xb1d5d564U, 0x9c4e4ed2U, 0x49a9a9e0U,
 117     0xd86c6cb4U, 0xac5656faU, 0xf3f4f407U, 0xcfeaea25U,
 118     0xca6565afU, 0xf47a7a8eU, 0x47aeaee9U, 0x10080818U,
 119     0x6fbabad5U, 0xf0787888U, 0x4a25256fU, 0x5c2e2e72U,
 120     0x381c1c24U, 0x57a6a6f1U, 0x73b4b4c7U, 0x97c6c651U,
 121     0xcbe8e823U, 0xa1dddd7cU, 0xe874749cU, 0x3e1f1f21U,
 122     0x964b4bddU, 0x61bdbddcU, 0x0d8b8b86U, 0x0f8a8a85U,
 123     0xe0707090U, 0x7c3e3e42U, 0x71b5b5c4U, 0xcc6666aaU,
 124     0x904848d8U, 0x06030305U, 0xf7f6f601U, 0x1c0e0e12U,
 125     0xc26161a3U, 0x6a35355fU, 0xae5757f9U, 0x69b9b9d0U,
 126     0x17868691U, 0x99c1c158U, 0x3a1d1d27U, 0x279e9eb9U,
```

```
 127     0xd9e1e138U, 0xebf8f813U, 0x2b9898b3U, 0x22111133U,
 128     0xd26969bbU, 0xa9d9d970U, 0x078e8e89U, 0x339494a7U,
 129     0x2d9b9bb6U, 0x3c1e1e22U, 0x15878792U, 0xc9e9e920U,
 130     0x87cece49U, 0xaa5555ffU, 0x50282878U, 0xa5dfdf7aU,
 131     0x038c8c8fU, 0x59a1a1f8U, 0x09898980U, 0x1a0d0d17U,
 132     0x65bfbfdaU, 0xd7e6e631U, 0x844242c6U, 0xd06868b8U,
 133     0x824141c3U, 0x299999b0U, 0x5a2d2d77U, 0x1e0f0f11U,
 134     0x7bb0b0cbU, 0xa85454fcU, 0x6dbbbbd6U, 0x2c16163aU,
 135 };
______unchanged_portion_omitted_
 742 /* EXPORT DELETE END */

 744 /* EXPORT DELETE START */

 742 int
 743 aes_init(void **cookie)
 744 {
 745         if ((*cookie = malloc(sizeof (keysched_t))) == NULL) {
 746                 return (-1);
 747         }
 748         return (0);
 749 }
______unchanged_portion_omitted_

1408 /* EXPORT DELETE END */
```

```
**********************************************************
    1330 Thu Jul 11 01:29:10 2013
new/usr/src/lib/crypt_modules/bsdbf/Makefile
first pass
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #
  25 # ident "%Z%%M% %I%     %E% SMI"
  26 #

  28 include $(SRC)/lib/Makefile.lib

  30 SUBDIRS=        $(MACH)
  31 $(BUILD64)SUBDIRS += $(MACH64)

  33 all :=          TARGET= all
  34 clean :=        TARGET= clean
  35 clobber :=      TARGET= clobber
  36 delete :=       TARGET= delete
  37 install :=      TARGET= install
  38 lint :=         TARGET= lint
  39 package :=      TARGET= package

  41 .KEEP_STATE:

  43 all clean clobber delete install lint package: $(SUBDIRS)

  45 _msg:

  47 catalog:

  49 $(SUBDIRS):     FRC
  50         @cd $@; pwd; $(MAKE) $(TARGET)

  52 FRC:

  54 # EXPORT DELETE START
  55 CRYPT_SRC: EXPORT_SRC
  56 EXPORT_SRC:
  57         $(RM) Makefile+ blowfish.c+
  58         sed -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
  59                 < blowfish.c > blowfish.c+
  60         $(MV) blowfish.c+ blowfish.c
  61         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
```



```
  62                 < Makefile > Makefile+
  63         $(MV) Makefile+ Makefile
  64         $(CHMOD) 444 Makefile blowfish.c

  66 # EXPORT DELETE END
```

```
**********************************************************
   23578 Thu Jul 11 01:29:11 2013
new/usr/src/lib/crypt_modules/bsdbf/blowfish.c
pass 2
**********************************************************
```

```
/*
 * Copyright 2002 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * The above Sun copyright is included due to changes made to this code
 * for US export control.  No changes to the algorithm implementations have
 * been made.
 */

#pragma ident   "%Z%%M% %I%     %E% SMI"

/* $OpenBSD: blowfish.c,v 1.16 2002/02/19 19:39:36 millert Exp $ */
/*
 * Blowfish block cipher for OpenBSD
 * Copyright 1997 Niels Provos <provos@physnet.uni-hamburg.de>
 * All rights reserved.
 *
 * Implementation advice by David Mazieres <dm@lcs.mit.edu>.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *      This product includes software developed by Niels Provos.
 * 4. The name of the author may not be used to endorse or promote products
 *    derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

/*
 * This code is derived from section 14.3 and the given source
 * in section V of Applied Cryptography, second edition.
 * Blowfish is an unpatented fast block cipher designed by
 * Bruce Schneier.
 */

#if 0
#include <stdio.h>              /* used for debugging */
#include <string.h>
#endif

#include <sys/types.h>
#include <blf.h>
```



```
#undef inline
#ifdef __GNUC__
#define inline __inline
#else                           /* !__GNUC__ */
#define inline
#endif                          /* !__GNUC__ */

/* Function for Feistel Networks */

#define F(s, x) ((((s)[        (((x)>>24)&0xFF)]  \
                 + (s)[0x100 + (((x)>>16)&0xFF)]) \
                 ^ (s)[0x200 + (((x)>> 8)&0xFF)]) \
                 + (s)[0x300 + ( (x)     &0xFF)])

#define BLFRND(s,p,i,j,n) (i ^= F(s,j) ^ (p)[n])

void
Blowfish_encipher(c, xl, xr)
        blf_ctx *c;
        uint32_t *xl;
        uint32_t *xr;
{
/* CRYPT DELETE START */
        uint32_t Xl;
        uint32_t Xr;
        uint32_t *s = c->S[0];
        uint32_t *p = c->P;

        Xl = *xl;
        Xr = *xr;

        Xl ^= p[0];
        BLFRND(s, p, Xr, Xl, 1); BLFRND(s, p, Xl, Xr, 2);
        BLFRND(s, p, Xr, Xl, 3); BLFRND(s, p, Xl, Xr, 4);
        BLFRND(s, p, Xr, Xl, 5); BLFRND(s, p, Xl, Xr, 6);
        BLFRND(s, p, Xr, Xl, 7); BLFRND(s, p, Xl, Xr, 8);
        BLFRND(s, p, Xr, Xl, 9); BLFRND(s, p, Xl, Xr, 10);
        BLFRND(s, p, Xr, Xl, 11); BLFRND(s, p, Xl, Xr, 12);
        BLFRND(s, p, Xr, Xl, 13); BLFRND(s, p, Xl, Xr, 14);
        BLFRND(s, p, Xr, Xl, 15); BLFRND(s, p, Xl, Xr, 16);

        *xl = Xr ^ p[17];
        *xr = Xl;
/* CRYPT DELETE END */
}

void
Blowfish_decipher(c, xl, xr)
        blf_ctx *c;
        uint32_t *xl;
        uint32_t *xr;
{
/* CRYPT DELETE START */
        uint32_t Xl;
        uint32_t Xr;
        uint32_t *s = c->S[0];
        uint32_t *p = c->P;

        Xl = *xl;
        Xr = *xr;

        Xl ^= p[17];
        BLFRND(s, p, Xr, Xl, 16); BLFRND(s, p, Xl, Xr, 15);
        BLFRND(s, p, Xr, Xl, 14); BLFRND(s, p, Xl, Xr, 13);
        BLFRND(s, p, Xr, Xl, 12); BLFRND(s, p, Xl, Xr, 11);
```

```
125         BLFRND(s, p, Xr, Xl, 10); BLFRND(s, p, Xl, Xr, 9);
126         BLFRND(s, p, Xr, Xl, 8); BLFRND(s, p, Xl, Xr, 7);
127         BLFRND(s, p, Xr, Xl, 6); BLFRND(s, p, Xl, Xr, 5);
128         BLFRND(s, p, Xr, Xl, 4); BLFRND(s, p, Xl, Xr, 3);
129         BLFRND(s, p, Xr, Xl, 2); BLFRND(s, p, Xl, Xr, 1);

131         *xl = Xr ^ p[0];
132         *xr = Xl;
136 /* CRYPT DELETE END */
133 }

135 void
136 Blowfish_initstate(c)
137         blf_ctx *c;
138 {
143 /* CRYPT DELETE START */

139 /* P-box and S-box tables initialized with digits of Pi */

141         const blf_ctx initstate =

143         { {
144                 {
145                         0xd1310ba6, 0x98dfb5ac, 0x2ffd72db, 0xd01adfb7,
146                         0xb8e1afed, 0x6a267e96, 0xba7c9045, 0xf12c7f99,
147                         0x24a19947, 0xb3916cf7, 0x0801f2e2, 0x858efc16,
148                         0x636920d8, 0x71574e69, 0xa458fea3, 0xf4933d7e,
149                         0x0d95748f, 0x728eb658, 0x718bcd58, 0x82154aee,
150                         0x7b54a41d, 0xc25a59b5, 0x9c30d539, 0x2af26013,
151                         0xc5d1b023, 0x286085f0, 0xca417918, 0xb8db38ef,
152                         0x8e79dcb0, 0x603a180e, 0x6c9e0e8b, 0xb01e8a3e,
153                         0xd71577c1, 0xbd314b27, 0x78af2fda, 0x55605c60,
154                         0xe65525f3, 0xaa55ab94, 0x57489862, 0x63e81440,
155                         0x55ca396a, 0x2aab10b6, 0xb4cc5c34, 0x1141e8ce,
156                         0xa15486af, 0x7c72e993, 0xb3ee1411, 0x636fbc2a,
157                         0x2ba9c55d, 0x741831f6, 0xce5c3e16, 0x9b87931e,
158                         0xafd6ba33, 0x6c24cf5c, 0x7a325381, 0x28958677,
159                         0x3b8f4898, 0x6b4bb9af, 0xc4bfe81b, 0x66282193,
160                         0x61d809cc, 0xfb21a991, 0x487cac60, 0x5dec8032,
161                         0xef845d5d, 0xe98575b1, 0xdc262302, 0xeb651b88,
162                         0x23893e81, 0xd396acc5, 0x0f6d6ff3, 0x83f44239,
163                         0x2e0b4482, 0xa4842004, 0x69c8f04a, 0x9e1f9b5e,
164                         0x21c66842, 0xf6e96c9a, 0x670c9c61, 0xabd388f0,
165                         0x6a51a0d2, 0xd8542f68, 0x960fa728, 0xab5133a3,
166                         0x6eef0b6c, 0x137a3be4, 0xba3bf050, 0x7efb2a98,
167                         0xa1f1651d, 0x39af0176, 0x66ca593e, 0x82430e88,
168                         0x8cee8619, 0x456f9fb4, 0x7d84a5c3, 0x3b8b5ebe,
169                         0xe06f75d8, 0x85c12073, 0x401a449f, 0x56c16aa6,
170                         0x4ed3aa62, 0x363f7706, 0x1bfedf72, 0x429b023d,
171                         0x37d0d724, 0xd00a1248, 0xdb0fead3, 0x49f1c09b,
172                         0x075372c9, 0x80991b7b, 0x25d479d8, 0xf6e8def7,
173                         0xe3fe501a, 0xb6794c3b, 0x976ce0bd, 0x04c006ba,
174                         0xc1a94fb6, 0x409f60c4, 0x5e5c9ec2, 0x196a2463,
175                         0x68fb6faf, 0x3e6c53b5, 0x1339b2eb, 0x3b52ec6f,
176                         0x6dfc511f, 0x9b30952c, 0xcc814544, 0xaf5ebd09,
177                         0xbee3d004, 0xde334afd, 0x660f2807, 0x192e4bb3,
178                         0xc0cba857, 0x45c8740f, 0xd20b5f39, 0xb9d3fbdb,
179                         0x5579c0bd, 0x1a60320a, 0xd6a100c6, 0x402c7279,
180                         0x679f25fe, 0xfb1fa3cc, 0x8ea5e9f8, 0xdb3222f8,
181                         0x3c7516df, 0xfd616b15, 0x2f501ec8, 0xad0552ab,
182                         0x323db5fa, 0xfd238760, 0x53317b48, 0x3e00df82,
183                         0x9e5c57bb, 0xca6f8ca0, 0x1a87562e, 0xdf1769db,
184                         0xd542a8f6, 0x287effc3, 0xac6732c6, 0x8c4f5573,
185                         0x695b27b0, 0xbbca58c8, 0xe1ffa35d, 0xb8f011a0,
186                         0x10fa3d98, 0xfd2183b8, 0x4afcb56c, 0x2dd1d35b,
187                         0x9a53e479, 0xb6f84565, 0xd28e49bc, 0x4bfb9790,
```



```
188                         0xe1ddf2da, 0xa4cb7e33, 0x62fb1341, 0xcee4c6e8,
189                         0xef20cada, 0x36774c01, 0xd07e9efe, 0x2bf11fb4,
190                         0x95dbda4d, 0xae909198, 0xeaad8e71, 0x6b93d5a0,
191                         0xd08ed1d0, 0xafc725e0, 0x8e3c5b2f, 0x8e7594b7,
192                         0x8ff6e2fb, 0xf2122b64, 0x8888b812, 0x900df01c,
193                         0x4fad5ea0, 0x688fc31c, 0xd1cff191, 0xb3a8c1ad,
194                         0x2f2f2218, 0xbe0e1777, 0xea752dfe, 0x8b021fa1,
195                         0xe5a0cc0f, 0xb56f74e8, 0x18acf3d6, 0xce89e299,
196                         0xb4a84fe0, 0xfd13e0b7, 0x7cc43b81, 0xd2ada8d9,
197                         0x165fa266, 0x80957705, 0x93cc7314, 0x211a1477,
198                         0xe6ad2065, 0x77b5fa86, 0xc75442f5, 0xfb9d35cf,
199                         0xebcdaf0c, 0x7b3e89a0, 0xd6411bd3, 0xae1e7e49,
200                         0x00250e2d, 0x2071b35e, 0x226800bb, 0x57b8e0af,
201                         0x2464369b, 0xf009b91e, 0x5563911d, 0x59dfa6aa,
202                         0x78c14389, 0xd95a537f, 0x207d5ba2, 0x02e5b9c5,
203                         0x83260376, 0x6295cfa9, 0x11c81968, 0x4e734a41,
204                         0xb3472dca, 0x7b14a94a, 0x1b510052, 0x9a532915,
205                         0xd60f573f, 0xbc9bc6e4, 0x2b60a476, 0x81e67400,
206                         0x08ba6fb5, 0x571be91f, 0xf296ec6b, 0x2a0dd915,
207                         0xb6636521, 0xe7b9f9b6, 0xff34052e, 0xc5855664,
208                 0x53b02d5d, 0xa99f8fa1, 0x08ba4799, 0x6e85076a},
209                 {
210                         0x4b7a70e9, 0xb5b32944, 0xdb75092e, 0xc4192623,
211                         0xad6ea6b0, 0x49a7df7d, 0x9cee60b8, 0x8fedb266,
212                         0xecaa8c71, 0x699a17ff, 0x5664526c, 0xc2b19ee1,
213                         0x193602a5, 0x75094c29, 0xa0591340, 0xe4183a3e,
214                         0x3f54989a, 0x5b429d65, 0x6b8fe4d6, 0x99f73fd6,
215                         0xa1d29c07, 0xefe830f5, 0x4d2d38e6, 0xf0255dc1,
216                         0x4cdd2086, 0x8470eb26, 0x6382e9c6, 0x021ecc5e,
217                         0x09686b3f, 0x3ebaefc9, 0x3c971814, 0x6b6a70a1,
218                         0x687f3584, 0x52a0e286, 0xb79c5305, 0xaa500737,
219                         0x3e07841c, 0x7fdeae5c, 0x8e7d44ec, 0x5716f2b8,
220                         0xb03ada37, 0xf0500c0d, 0xf01c1f04, 0x0200b3ff,
221                         0xae0cf51a, 0x3cb574b2, 0x25837a58, 0xdc0921bd,
222                         0xd19113f9, 0x7ca92ff6, 0x94324773, 0x22f54701,
223                         0x3ae5e581, 0x37c2dadc, 0xc8b57634, 0x9af3dda7,
224                         0xa9446146, 0x0fd0030e, 0xecc8c73e, 0xa4751e41,
225                         0xe238cd99, 0x3bea0e2f, 0x3280bba1, 0x183eb331,
226                         0x4e548b38, 0x4f6db908, 0x6f420d03, 0xf60a04bf,
227                         0x2cb81290, 0x24977c79, 0x5679b072, 0xbcaf89af,
228                         0xde9a771f, 0xd9930810, 0xb38bae12, 0xdccf3f2e,
229                         0x5512721f, 0x2e6b7124, 0x501adde6, 0x9f84cd87,
230                         0x7a584718, 0x7408da17, 0xbc9f9abc, 0xe94b7d8c,
231                         0xec7aec3a, 0xdb851dfa, 0x63094366, 0xc464c3d2,
232                         0xef1c1847, 0x3215d908, 0xdd433b37, 0x24c2ba16,
233                         0x12a14d43, 0x2a65c451, 0x50940002, 0x133ae4dd,
234                         0x71dff89e, 0x10314e55, 0x81ac77d6, 0x5f11199b,
235                         0x043556f1, 0xd7a3c76b, 0x3c11183b, 0x5924a509,
236                         0xf28fe6ed, 0x97f1fbfa, 0x9ebabf2c, 0x1e153c6e,
237                         0x86e34570, 0xeae96fb1, 0x860e5e0a, 0x5a3e2ab3,
238                         0x771fe71c, 0x4e3d06fa, 0x2965dcb9, 0x99e71d0f,
239                         0x803e89d6, 0x5266c825, 0x2e4cc978, 0x9c10b36a,
240                         0xc6150eba, 0x94e2ea78, 0xa5fc3c53, 0x1e0a2df4,
241                         0xf2f74ea7, 0x361d2b3d, 0x1939260f, 0x19c27960,
242                         0x5223a708, 0xf71312b6, 0xebadfe6e, 0xeac31f66,
243                         0xe3bc4595, 0xa67bc883, 0xb17f37d1, 0x018cff28,
244                         0xc332ddef, 0xbe6c5aa5, 0x65582185, 0x68ab9802,
245                         0xeecea50f, 0xdb2f953b, 0x2aef7dad, 0x5b6e2f84,
246                         0x1521b628, 0x29076170, 0xecdd4775, 0x619f1510,
247                         0x13cca830, 0xeb61bd96, 0x0334fe1e, 0xaa0363cf,
248                         0xb5735c90, 0x4c70a239, 0xd59e9e0b, 0xcbaade14,
249                         0xeecc86bc, 0x60622ca7, 0x9cab5cab, 0xb2f3846e,
250                         0x648b1eaf, 0x19bdf0ca, 0xa02369b9, 0x655abb50,
251                         0x40685a32, 0x3c2ab4b3, 0x319ee9d5, 0xc021b8f7,
252                         0x9b540b19, 0x875fa099, 0x95f7997e, 0x623d7da8,
253                         0xf837889a, 0x97e32d77, 0x11ed935f, 0x16681281,
```

```
                        0x0e358829, 0xc7e61fd6, 0x96dedfa1, 0x7858ba99,
                        0x57f584a5, 0x1b227263, 0x9b83c3ff, 0x1ac24696,
                        0xcdb30aeb, 0x532e3054, 0x8fd948e4, 0x6dbc3128,
                        0x58ebf2ef, 0x34c6ffea, 0xfe28ed61, 0xee7c3c73,
                        0x5d4a14d9, 0xe864b7e3, 0x42105d14, 0x203e13e0,
                        0x45eee2b6, 0xa3aaabea, 0xdb6c4f15, 0xfacb4fd0,
                        0xc742f442, 0xef6abbb5, 0x654f3b1d, 0x41cd2105,
                        0xd81e799e, 0x86854dc7, 0xe44b476a, 0x3d816250,
                        0xcf62a1f2, 0x5b8d2646, 0xfc8883a0, 0xc1c7b6a3,
                        0x7f1524c3, 0x69cb7492, 0x47848a0b, 0x5692b285,
                        0x095bbf00, 0xad19489d, 0x1462b174, 0x23820e00,
                        0x58428d2a, 0x0c55f5ea, 0x1dadf43e, 0x233f7061,
                        0x3372f092, 0x8d937e41, 0xd65fecf1, 0x6c223bdb,
                        0x7cde3759, 0xcbee7460, 0x4085f2a7, 0xce77326e,
                        0xa6078084, 0x19f8509e, 0xe8efd855, 0x61d99735,
                        0xa969a7aa, 0xc50c06c2, 0x5a04abfc, 0x800bcadc,
                        0x9e447a2e, 0xc3453484, 0xfdd56705, 0x0e1e9ec9,
                        0xdb73dbd3, 0x105588cd, 0x675fda79, 0xe3674340,
                        0xc5c43465, 0x713e38d8, 0x3d28f89e, 0xf16dff20,
                0x153e21e7, 0x8fb03d4a, 0xe6e39f2b, 0xdb83adf7},
                {
                        0xe93d5a68, 0x948140f7, 0xf64c261c, 0x94692934,
                        0x411520f7, 0x7602d4f7, 0xbcf46b2e, 0xd4a20068,
                        0xd4082471, 0x3320f46a, 0x43b7d4b7, 0x500061af,
                        0x1e39f62e, 0x97244546, 0x14214f74, 0xbf8b8840,
                        0x4d95fc1d, 0x96b591af, 0x70f4ddd3, 0x66a02f45,
                        0xbfbc09ec, 0x03bd9785, 0x7fac6dd0, 0x31cb8504,
                        0x96eb27b3, 0x55fd3941, 0xda2547e6, 0xabca0a9a,
                        0x28507825, 0x530429f4, 0x0a2c86da, 0xe9b66dfb,
                        0x68dc1462, 0xd7486900, 0x680ec0a4, 0x27a18dee,
                        0x4f3ffea2, 0xe887ad8c, 0xb58ce006, 0x7af4d6b6,
                        0xaace1e7c, 0xd3375fec, 0xce78a399, 0x406b2a42,
                        0x20fe9e35, 0xd9f385b9, 0xee39d7ab, 0x3b124e8b,
                        0x1dc9faf7, 0x4b6d1856, 0x26a36631, 0xeae397b2,
                        0x3a6efa74, 0xdd5b4332, 0x6841e7f7, 0xca7820fb,
                        0xfb0af54e, 0xd8feb397, 0x454056ac, 0xba489527,
                        0x55533a3a, 0x20838d87, 0xfe6ba9b7, 0xd096954b,
                        0x55a867bc, 0xa1159a58, 0xcca92963, 0x99e1db33,
                        0xa62a4a56, 0x3f3125f9, 0x5ef47e1c, 0x9029317c,
                        0xfdf8e802, 0x04272f70, 0x80bb155c, 0x05282ce3,
                        0x95c11548, 0xe4c66d22, 0x48c1133f, 0xc70f86dc,
                        0x07f9c9ee, 0x41041f0f, 0x404779a4, 0x5d886e17,
                        0x325f51eb, 0xd59bc0d1, 0xf2bcc18f, 0x41113564,
                        0x257b7834, 0x602a9c60, 0xdff8e8a3, 0x1f636c1b,
                        0x0e12b4c2, 0x02e1329e, 0xaf664fd1, 0xcad18115,
                        0x6b2395e0, 0x333e92e1, 0x3b240b62, 0xeebeb922,
                        0x85b2a20e, 0xe6ba0d99, 0xde720c8c, 0x2da2f728,
                        0xd0127845, 0x95b794fd, 0x647d0862, 0xe7ccf5f0,
                        0x5449a36f, 0x877d48fa, 0xc39dfd27, 0xf33e8d1e,
                        0x0a476341, 0x992eff74, 0x3a6f6eab, 0xf4f8fd37,
                        0xa812dc60, 0xa1ebddf8, 0x991be14c, 0xdb6e6b0d,
                        0xc67b5510, 0x6d672c37, 0x2765d43b, 0xdcd0e804,
                        0xf1290dc7, 0xcc00ffa3, 0xb5390f92, 0x690fed0b,
                        0x667b9ffb, 0xcedb7d9c, 0xa091cf0b, 0xd9155ea3,
                        0xbb132f88, 0x515bad24, 0x7b9479bf, 0x763bd6eb,
                        0x37392eb3, 0xcc115979, 0x8026e297, 0xf42e312d,
                        0x6842ada7, 0xc66a2b3b, 0x12754ccc, 0x782ef11c,
                        0x6a124237, 0xb79251e7, 0x06a1bbe6, 0x4bfb6350,
                        0x1a6b1018, 0x11caedfa, 0x3d25bdd8, 0xe2e1c3c9,
                        0x44421659, 0x0a121386, 0xd90cec6e, 0xd5abea2a,
                        0x64af674e, 0xda86a85f, 0xbebfe988, 0x64e4c3fe,
                        0x9dbc8057, 0xf0f7c086, 0x60787bf8, 0x6003604d,
                        0xd1fd8346, 0xf6381fb0, 0x7745ae04, 0xd736fccc,
                        0x83426b33, 0xf01eab71, 0xb0804187, 0x3c005e5f,
                        0x77a057be, 0xbde8ae24, 0x55464299, 0xbf582e61,
                        0x4e58f48f, 0xf2ddfda2, 0xf474ef38, 0x8789bdc2,
```



```
                        0x5366f9c3, 0xc8b38e74, 0xb475f255, 0x46fcd9b9,
                        0x7aeb2661, 0x8b1ddf84, 0x846a0e79, 0x915f95e2,
                        0x466e598e, 0x20b45770, 0x8cd55591, 0xc902de4c,
                        0xb90bace1, 0xbb8205d0, 0x11a86248, 0x7574a99e,
                        0xb77f19b6, 0xe0a9dc09, 0x662d09a1, 0xc4324633,
                        0xe85a1f02, 0x09f0be8c, 0x4a99a025, 0x1d6efe10,
                        0x1ab93d1d, 0x0ba5a4df, 0xa186f20f, 0x2868f169,
                        0xdcb7da83, 0x573906fe, 0xa1e2ce9b, 0x4fcd7f52,
                        0x50115e01, 0xa70683fa, 0xa002b5c4, 0x0de6d027,
                        0x9af88c27, 0x773f8641, 0xc3604c06, 0x61a806b5,
                        0xf0177a28, 0xc0f586e0, 0x006058aa, 0x30dc7d62,
                        0x11e69ed7, 0x2338ea63, 0x53c2dd94, 0xc2c21634,
                        0xbbcbee56, 0x90bcb6de, 0xebfc7da1, 0xce591d76,
                        0x6f05e409, 0x4b7c0188, 0x39720a3d, 0x7c927c24,
                        0x86e3725f, 0x724d9db9, 0x1ac15bb4, 0xd39eb8fc,
                        0xed545578, 0x08fca5b5, 0xd83d7cd3, 0x4dad0fc4,
                        0x1e50ef5e, 0xb161e6f8, 0xa28514d9, 0x6c51133c,
                        0x6fd5c7e7, 0x56e14ec4, 0x362abfce, 0xddc6c837,
                0xd79a3234, 0x92638212, 0x670efa8e, 0x406000e0},
                {
                        0x3a39ce37, 0xd3faf5cf, 0xabc27737, 0x5ac52d1b,
                        0x5cb0679e, 0x4fa33742, 0xd3822740, 0x99bc9bbe,
                        0xd5118e9d, 0xbf0f7315, 0xd62d1c7e, 0xc700c47b,
                        0xb78c1b6b, 0x21a19045, 0xb26eb1be, 0x6a366eb4,
                        0x5748ab2f, 0xbc946e79, 0xc6a376d2, 0x6549c2c8,
                        0x530ff8ee, 0x468dde7d, 0xd5730a1d, 0x4cd04dc6,
                        0x2939bbdb, 0xa9ba4650, 0xac9526e8, 0xbe5ee304,
                        0xa1fad5f0, 0x6a2d519a, 0x63ef8ce2, 0x9a86ee22,
                        0xc089c2b8, 0x43242ef6, 0xa51e03aa, 0x9cf2d0a4,
                        0x83c061ba, 0x9be96a4d, 0x8fe51550, 0xba645bd6,
                        0x2826a2f9, 0xa73a3ae1, 0x4ba99586, 0xef5562e9,
                        0xc72fefd3, 0xf752f7da, 0x3f046f69, 0x77fa0a59,
                        0x80e4a915, 0x87b08601, 0x9b09e6ad, 0x3b3ee593,
                        0xe990fd5a, 0x9e34d797, 0x2cf0b7d9, 0x022b8b51,
                        0x96d5ac3a, 0x017da67d, 0xd1cf3ed6, 0x7c7d2d28,
                        0x1f9f25cf, 0xadf2b89b, 0x5ad6b472, 0x5a88f54c,
                        0xe029ac71, 0xe019a5e6, 0x47b0acfd, 0xed93fa9b,
                        0xe8d3c48d, 0x283b57cc, 0xf8d56629, 0x79132e28,
                        0x785f0191, 0xed756055, 0xf7960e44, 0xe3d35e8c,
                        0x15056dd4, 0x88f46dba, 0x03a16125, 0x0564f0bd,
                        0xc3eb9e15, 0x3c9057a2, 0x97271aec, 0xa93a072a,
                        0x1b3f6d9b, 0x1e6321f5, 0xf59c66fb, 0x26dcf319,
                        0x7533d928, 0xb155fdf5, 0x03563482, 0x8aba3cbb,
                        0x28517711, 0xc20ad9f8, 0xabcc5167, 0xccad925f,
                        0x4de81751, 0x3830dc8e, 0x379d5862, 0x9320f991,
                        0xea7a90c2, 0xfb3e7bce, 0x5121ce64, 0x774fbe32,
                        0xa8b6e37e, 0xc3293d46, 0x48de5369, 0x6413e680,
                        0xa2ae0810, 0xdd6db224, 0x69852dfd, 0x09072166,
                        0xb39a460a, 0x6445c0dd, 0x586cdecf, 0x1c20c8ae,
                        0x5bbef7dd, 0x1b588d40, 0xccd2017f, 0x6bb4e3bb,
                        0xdda26a7e, 0x3a59ff45, 0x3e350a44, 0xbcb4cdd5,
                        0x72eacea8, 0xfa6484bb, 0x8d6612ae, 0xbf3c6f47,
                        0xd29be463, 0x542f5d9e, 0xaec2771b, 0xf64e6370,
                        0x740e0d8d, 0xe75b1357, 0xf8721671, 0xaf537d5d,
                        0x4040cb08, 0x4eb4e2cc, 0x34d2466a, 0x0115af84,
                        0xe1b00428, 0x95983a1d, 0x06b89fb4, 0xce6ea048,
                        0x6f3f3b82, 0x3520ab82, 0x011a1d4b, 0x277227f8,
                        0x611560b1, 0xe7933fdc, 0xbb3a792b, 0x344525bd,
                        0xa08839e1, 0x51ce794b, 0x2f32c9b7, 0xa01fbac9,
                        0xe01cc87e, 0xbcc7d1f6, 0xcf0111c3, 0xa1e8aac7,
                        0x1a908749, 0xd44fbd9a, 0xd0dadecb, 0xd50ada38,
                        0x0339c32a, 0xc6913667, 0x8df9317c, 0xe0b12b4f,
                        0xf79e59b7, 0x43f5bb3a, 0xf2d519ff, 0x27d9459c,
                        0xbf97222c, 0x15e6fc2a, 0x0f91fc71, 0x9b941525,
                        0xfae59361, 0xceb69ceb, 0xc2a86459, 0x12baa8d1,
                        0xb6c1075e, 0xe3056a0c, 0x10d25065, 0xcb03a442,
```

```
386                         0xe0ec6e0e, 0x1698db3b, 0x4c98a0be, 0x3278e964,
387                         0x9f1f9532, 0xe0d392df, 0xd3a0342b, 0x8971f21e,
388                         0x1b0a7441, 0x4ba3348c, 0xc5be7120, 0xc37632d8,
389                         0xdf359f8d, 0x9b992f2e, 0xe60b6f47, 0x0fe3f11d,
390                         0xe54cda54, 0x1edad891, 0xce6279cf, 0xcd3e7e6f,
391                         0x1618b166, 0xfd2c1d05, 0x848fd2c5, 0xf6fb2299,
392                         0xf523f357, 0xa6327623, 0x93a83531, 0x56cccd02,
393                         0xacf08162, 0x5a75ebb5, 0x6e163697, 0x88d273cc,
394                         0xde966292, 0x81b949d0, 0x4c50901b, 0x71c65614,
395                         0xe6c6c7bd, 0x327a140a, 0x45e1d006, 0xc3f27b9a,
396                         0xc9aa53fd, 0x62a80f00, 0xbb25bfe2, 0x35bdd2f6,
397                         0x71126905, 0xb2040222, 0xb6cbcf7c, 0xcd769c2b,
398                         0x53113ec0, 0x1640e3d3, 0x38abbd60, 0x2547adf0,
399                         0xba38209c, 0xf746ce76, 0x77afa1c5, 0x20756060,
400                         0x85cbfe4e, 0x8ae88dd8, 0x7aaaf9b0, 0x4cf9aa7e,
401                         0x1948c25c, 0x02fb8a8c, 0x01c36ae4, 0xd6ebe1f9,
402                         0x90d4f869, 0xa65cdea0, 0x3f09252d, 0xc208e69f,
403                 0xb74e6132, 0xce77e25b, 0x578fdfe3, 0x3ac372e6}
404         },
405         {
406                 0x243f6a88, 0x85a308d3, 0x13198a2e, 0x03707344,
407                 0xa4093822, 0x299f31d0, 0x082efa98, 0xec4e6c89,
408                 0x452821e6, 0x38d01377, 0xbe5466cf, 0x34e90c6c,
409                 0xc0ac29b7, 0xc97c50dd, 0x3f84d5b5, 0xb5470917,
410                 0x9216d5d9, 0x8979fb1b
411         } };

413         *c = initstate;

421 /* CRYPT DELETE END */
414 }

416 uint32_t
417 Blowfish_stream2word(const uint8_t *data, uint16_t databytes, uint16_t *current)
418 {
419         uint8_t i;
420         uint16_t j;
421         uint32_t temp;

423         temp = 0x00000000;
432 /* CRYPT DELETE START */
424         j = *current;

426         for (i = 0; i < 4; i++, j++) {
427                 if (j >= databytes)
428                         j = 0;
429                 temp = (temp << 8) | data[j];
430         }

432         *current = j;
442 /* CRYPT DELETE END */
433         return temp;
434 }

436 void
437 Blowfish_expand0state(blf_ctx *c, const uint8_t *key, uint16_t keybytes)
438 {
449 /* CRYPT DELETE START */
439         uint16_t i;
440         uint16_t j;
441         uint16_t k;
442         uint32_t temp;
443         uint32_t datal;
444         uint32_t datar;

446         j = 0;
```



```
447         for (i = 0; i < BLF_N + 2; i++) {
448                 /* Extract 4 int8 to 1 int32 from keystream */
449                 temp = Blowfish_stream2word(key, keybytes, &j);
450                 c->P[i] = c->P[i] ^ temp;
451         }

453         j = 0;
454         datal = 0x00000000;
455         datar = 0x00000000;
456         for (i = 0; i < BLF_N + 2; i += 2) {
457                 Blowfish_encipher(c, &datal, &datar);

459                 c->P[i] = datal;
460                 c->P[i + 1] = datar;
461         }

463         for (i = 0; i < 4; i++) {
464                 for (k = 0; k < 256; k += 2) {
465                         Blowfish_encipher(c, &datal, &datar);

467                         c->S[i][k] = datal;
468                         c->S[i][k + 1] = datar;
469                 }
470         }
482 /* CRYPT DELETE END */
471 }


474 void
475 Blowfish_expandstate(blf_ctx *c, const uint8_t *data, uint16_t databytes,
476                         const uint8_t *key, uint16_t keybytes)
477 {
490 /* CRYPT DELETE START */
478         uint16_t i;
479         uint16_t j;
480         uint16_t k;
481         uint32_t temp;
482         uint32_t datal;
483         uint32_t datar;

485         j = 0;
486         for (i = 0; i < BLF_N + 2; i++) {
487                 /* Extract 4 int8 to 1 int32 from keystream */
488                 temp = Blowfish_stream2word(key, keybytes, &j);
489                 c->P[i] = c->P[i] ^ temp;
490         }

492         j = 0;
493         datal = 0x00000000;
494         datar = 0x00000000;
495         for (i = 0; i < BLF_N + 2; i += 2) {
496                 datal ^= Blowfish_stream2word(data, databytes, &j);
497                 datar ^= Blowfish_stream2word(data, databytes, &j);
498                 Blowfish_encipher(c, &datal, &datar);

500                 c->P[i] = datal;
501                 c->P[i + 1] = datar;
502         }

504         for (i = 0; i < 4; i++) {
505                 for (k = 0; k < 256; k += 2) {
506                         datal ^= Blowfish_stream2word(data, databytes, &j);
507                         datar ^= Blowfish_stream2word(data, databytes, &j);
508                         Blowfish_encipher(c, &datal, &datar);

510                         c->S[i][k] = datal;
```

```
511                         c->S[i][k + 1] = datar;
512                 }
513         }

528 /* CRYPT DELETE END */
514 }

516 void
517 blf_key(blf_ctx *c, const uint8_t *k, uint16_t len)
518 {
534 /* CRYPT DELETE START */
519         /* Initialize S-boxes and subkeys with Pi */
520         Blowfish_initstate(c);

522         /* Transform S-boxes and subkeys with key */
523         Blowfish_expand0state(c, k, len);
540 /* CRYPT DELETE END */
524 }

526 void
527 blf_enc(blf_ctx *c, uint32_t *data, uint16_t blocks)
528 {
546 /* CRYPT DELETE START */
529         uint32_t *d;
530         uint16_t i;

532         d = data;
533         for (i = 0; i < blocks; i++) {
534                 Blowfish_encipher(c, d, d + 1);
535                 d += 2;
536         }
555 /* CRYPT DELETE END */
537 }

539 void
540 blf_dec(blf_ctx *c, uint32_t *data, uint16_t blocks)
541 {
561 /* CRYPT DELETE START */
542         uint32_t *d;
543         uint16_t i;

545         d = data;
546         for (i = 0; i < blocks; i++) {
547                 Blowfish_decipher(c, d, d + 1);
548                 d += 2;
549         }
570 /* CRYPT DELETE END */
550 }

552 void
553 blf_ecb_encrypt(blf_ctx *c, uint8_t *data, uint32_t len)
554 {
576 /* CRYPT DELETE START */
555         uint32_t l, r;
556         uint32_t i;

558         for (i = 0; i < len; i += 8) {
559                 l = data[0] << 24 | data[1] << 16 | data[2] << 8 | data[3];
560                 r = data[4] << 24 | data[5] << 16 | data[6] << 8 | data[7];
561                 Blowfish_encipher(c, &l, &r);
562                 data[0] = l >> 24 & 0xff;
563                 data[1] = l >> 16 & 0xff;
564                 data[2] = l >> 8 & 0xff;
565                 data[3] = l & 0xff;
566                 data[4] = r >> 24 & 0xff;
567                 data[5] = r >> 16 & 0xff;
```



```
568                 data[6] = r >> 8 & 0xff;
569                 data[7] = r & 0xff;
570                 data += 8;
571         }
594 /* CRYPT DELETE END */
572 }

574 void
575 blf_ecb_decrypt(blf_ctx *c, uint8_t *data, uint32_t len)
576 {
600 /* CRYPT DELETE START */
577         uint32_t l, r;
578         uint32_t i;

580         for (i = 0; i < len; i += 8) {
581                 l = data[0] << 24 | data[1] << 16 | data[2] << 8 | data[3];
582                 r = data[4] << 24 | data[5] << 16 | data[6] << 8 | data[7];
583                 Blowfish_decipher(c, &l, &r);
584                 data[0] = l >> 24 & 0xff;
585                 data[1] = l >> 16 & 0xff;
586                 data[2] = l >> 8 & 0xff;
587                 data[3] = l & 0xff;
588                 data[4] = r >> 24 & 0xff;
589                 data[5] = r >> 16 & 0xff;
590                 data[6] = r >> 8 & 0xff;
591                 data[7] = r & 0xff;
592                 data += 8;
593         }
618 /* CRYPT DELETE END */
594 }

596 void
597 blf_cbc_encrypt(blf_ctx *c, uint8_t *iv, uint8_t *data, uint32_t len)
598 {
624 /* CRYPT DELETE START */
599         uint32_t l, r;
600         uint32_t i, j;

602         for (i = 0; i < len; i += 8) {
603                 for (j = 0; j < 8; j++)
604                         data[j] ^= iv[j];
605                 l = data[0] << 24 | data[1] << 16 | data[2] << 8 | data[3];
606                 r = data[4] << 24 | data[5] << 16 | data[6] << 8 | data[7];
607                 Blowfish_encipher(c, &l, &r);
608                 data[0] = l >> 24 & 0xff;
609                 data[1] = l >> 16 & 0xff;
610                 data[2] = l >> 8 & 0xff;
611                 data[3] = l & 0xff;
612                 data[4] = r >> 24 & 0xff;
613                 data[5] = r >> 16 & 0xff;
614                 data[6] = r >> 8 & 0xff;
615                 data[7] = r & 0xff;
616                 iv = data;
617                 data += 8;
618         }
645 /* CRYPT DELETE END */
619 }

621 void
622 blf_cbc_decrypt(blf_ctx *c, uint8_t *iva, uint8_t *data, uint32_t len)
623 {
651 /* CRYPT DELETE START */
624         uint32_t l, r;
625         uint8_t *iv;
626         uint32_t i, j;
```

```
628         iv = data + len - 16;
629         data = data + len - 8;
630         for (i = len - 8; i >= 8; i -= 8) {
631                 l = data[0] << 24 | data[1] << 16 | data[2] << 8 | data[3];
632                 r = data[4] << 24 | data[5] << 16 | data[6] << 8 | data[7];
633                 Blowfish_decipher(c, &l, &r);
634                 data[0] = l >> 24 & 0xff;
635                 data[1] = l >> 16 & 0xff;
636                 data[2] = l >> 8 & 0xff;
637                 data[3] = l & 0xff;
638                 data[4] = r >> 24 & 0xff;
639                 data[5] = r >> 16 & 0xff;
640                 data[6] = r >> 8 & 0xff;
641                 data[7] = r & 0xff;
642                 for (j = 0; j < 8; j++)
643                         data[j] ^= iv[j];
644                 iv -= 8;
645                 data -= 8;
646         }
647         l = data[0] << 24 | data[1] << 16 | data[2] << 8 | data[3];
648         r = data[4] << 24 | data[5] << 16 | data[6] << 8 | data[7];
649         Blowfish_decipher(c, &l, &r);
650         data[0] = l >> 24 & 0xff;
651         data[1] = l >> 16 & 0xff;
652         data[2] = l >> 8 & 0xff;
653         data[3] = l & 0xff;
654         data[4] = r >> 24 & 0xff;
655         data[5] = r >> 16 & 0xff;
656         data[6] = r >> 8 & 0xff;
657         data[7] = r & 0xff;
658         for (j = 0; j < 8; j++)
659                 data[j] ^= iva[j];
688 /* CRYPT DELETE END */
660 }

691 /* CRYPT DELETE START */
662 #if 0
663 void
664 report(uint32_t data[], uint16_t len)
665 {
666         uint16_t i;
667         for (i = 0; i < len; i += 2)
668                 printf("Block %0hd: %08lx %08lx.\n",
669                     i / 2, data[i], data[i + 1]);
670 }
_____unchanged_portion_omitted_
704 #endif
735 /* CRYPT DELETE END */
```

```
**********************************************************
    2480 Thu Jul 11 01:29:12 2013
new/usr/src/lib/gss_mechs/mech_dh/backend/Makefile
first pass
**********************************************************
```

```
   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 2008 Sun Microsystems, Inc.  All rights reserved.
  24 # Use is subject to license terms.
  25 #
  26 # lib/gss_mechs/mech_dh/backend/Makefile
  27 #
  28 PROTOCOL_DIR = mech
  29 SUBDIRS = $(MACH) $(BUILD64) $(MACH64)

  31 PROTO_FILE = dhmech_prot.x
  32 DERIVED_FILES = mech/dhmech_prot.h mech/xdr_token.c
  33 CLEANFILES += $(DERIVED_FILES)

  35 # include library definitions
  36 include ../../../Makefile.lib

  38 TEXT_DOMAIN =   SUNW_OST_NETRPC
  39 POFILES =      generic.po
  40 POFILE = mech_dh.po

  42 SED=    sed
  43 GREP=   grep

  45 all :=          TARGET= all
  46 clean :=        TARGET= clean
  47 clobber :=      TARGET= clobber
  48 delete :=       TARGET= delete
  49 install :=      TARGET= install
  50 lint :=         TARGET= lint
  51 catalog :=      TARGET= catalog
  52 package :=      TARGET= package
  53 _msg :=         TARGET= _msg

  55 .KEEP_STATE:
  56 all:            $(DERIVED_FILES) .WAIT $(SUBDIRS)

  58 install:        $(DERIVED_FILES) .WAIT $(SUBDIRS)

  60 clean clobber delete lint catalog package:      $(SUBDIRS)
```



```
  62 #
  63 # Rules for building the derived files
  64 #
  65 mech/xdr_token.c: $(PROTOCOL_DIR)/dhmech_prot.x
  66         $(RPCGEN) -c $(PROTOCOL_DIR)/dhmech_prot.x |\
  67         $(SED) -e 's!$(PROTOCOL_DIR)/dhmech_prot.h!dhmech_prot.h!' > $@

  69 mech/dhmech_prot.h: mech/dhmech_prot.x
  70         $(RPCGEN) -h $(PROTOCOL_DIR)/dhmech_prot.x > $@

  72 # include library targets
  73 include ../../../Makefile.targ

  75 # EXPORT DELETE START
  76 # Special target to clean up the source tree for export distribution
  77 # Warning: This target changes the source tree
  78 EXPORT_SRC:
  79         $(RM) Makefile+ Makefile.com+ mech/crypto.c+ mech/dhmech.c+
  80         $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d"    \
  81                         < Makefile > Makefile+
  82         $(MV) Makefile+ Makefile
  83         $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d"    \
  84                         < Makefile.com > Makefile.com+
  85         $(MV) Makefile.com+ Makefile.com
  86         $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  87                 < mech/crypto.c > mech/crypto.c+
  88         $(MV) mech/crypto.c+ mech/crypto.c
  89         $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  90                 < mech/dhmech.c > mech/dhmech.c+
  91         $(MV) mech/dhmech.c+ mech/dhmech.c
  92         $(CHMOD) 444 Makefile Makefile.com mech/crypto.c mech/dhmech.c

  94 # CRYPT DELETE START
  95 CRYPT_SRC:
  96         $(RM) Makefile+ Makefile.com+ mech/dhmech.c+
  97         $(SED) -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d"      \
  98                         < Makefile                                      \
  99            | $(SED) -e "/EXPORT DELETE/d"                               \
 100                         > Makefile+
 101         $(MV) Makefile+ Makefile
 102         $(SED) -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d"      \
 103                         < Makefile.com                                  \
 104            | $(SED) -e "/EXPORT DELETE/d"                               \
 105                         > Makefile.com+
 106         $(MV) Makefile.com+ Makefile.com
 107         $(SED) -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
 108                         < mech/dhmech.c > mech/dhmech.c+
 109         $(MV) mech/dhmech.c+ mech/dhmech.c
 110         $(CHMOD) 444 Makefile Makefile.com mech/dhmech.c

 112 # CRYPT DELETE END
 113 # EXPORT DELETE END

  75 _msg: $(MSGDOMAIN) $(POFILE)
  76         $(RM) $(MSGDOMAIN)/$(POFILE)
  77         $(CP) $(POFILE) $(MSGDOMAIN)

  79 $(POFILE): $(DERIVED_FILES) .WAIT $(POFILES)
  80         $(RM) $@
  81         $(CAT) $(POFILES) > $@

  83 generic.po:
  84         $(RM) messages.po
  85         $(XGETTEXT) $(XGETFLAGS) `$(GREP) -l gettext mech/*.[ch]`
  86         $(SED) "/^domain/d" messages.po > $@
  87         $(RM) messages.po
```

```
 89 $(MSGDOMAIN):
 90         $(INS.dir)

 92 $(MACH) $(MACH64):      FRC
 93         @cd $@; pwd; $(MAKE) $(TARGET)

 95 FRC:
```

```
**********************************************************
   15670 Thu Jul 11 01:29:12 2013
new/usr/src/lib/gss_mechs/mech_dh/backend/mech/crypto.c
first pass
**********************************************************
______unchanged_portion_omitted_


 84 /* EXPORT DELETE START */

 84 /*
 85  * Des [en/de]crypt buffer, buf of length, len for each key provided using
 86  * an CBC initialization vector ivec.
 87  * If the mode is encrypt we will use the following pattern if the number
 88  * of keys is odd
 89  * encrypt(buf, k[0]), decrypt(buf, k[1]), encrypt(buf, k[2])
 90  *      decrypt(buf, k[4]) ... encrypt(buf, k[keynum - 1])
 91  * If we have an even number of keys and additional encryption will be
 92  * done with the first key, i.e., ecrypt(buf, k[0]);
 93  * In each [en/de]cription above we will used the passed in CBC initialization
 94  * vector. The new initialization vector will be the vector return from the
 95  * last encryption.
 96  *
 97  * In the decryption case we reverse the proccess. Note in this case
 98  * the return ivec will be from the first decryption.
 99  */

101 static int
102 __desN_crypt(des_block keys[], int keynum, char *buf, unsigned int len,
103     unsigned int mode, char *ivec)
104 {
105         /* Get the direction of ciphering */
106         unsigned int m = mode & (DES_ENCRYPT | DES_DECRYPT);
107         /* Get the remaining flags from mode */
108         unsigned int flags = mode & ~(DES_ENCRYPT | DES_DECRYPT);
109         des_block svec, dvec;
110         int i, j, stat;

112         /* Do we have at least one key */
113         if (keynum < 1)
114                 return (DESERR_BADPARAM);

116         /* Save the passed in ivec */
117         memcpy(svec.c, ivec, sizeof (des_block));

119         /* For  each key do the appropriate cipher */
120         for (i = 0; i < keynum; i++) {
121                 j = (mode & DES_DECRYPT) ? keynum - 1 - i : i;
122                 stat = cbc_crypt(keys[j].c, buf, len, m | flags, ivec);
123                 if (mode & DES_DECRYPT && i == 0)
124                         memcpy(dvec.c, ivec, sizeof (des_block));

126                 if (DES_FAILED(stat))
127                         return (stat);

129                 m = (m == DES_ENCRYPT ? DES_DECRYPT : DES_ENCRYPT);

131                 if ((mode & DES_DECRYPT) || i != keynum - 1 || i%2)
132                         memcpy(ivec, svec.c, sizeof (des_block));
133         }

135         /*
136          * If we have an even number of keys then do an extra round of
137          * [en/de]cryption with the first key.
138          */
139         if (keynum % 2 == 0)
```



```
140                 stat = cbc_crypt(keys[0].c, buf, len, mode, ivec);

142         /* If were decrypting ivec is set from first decryption */
143         if (mode & DES_DECRYPT)
144                 memcpy(ivec, dvec.c, sizeof (des_block));

146         return (stat);
147 }

151 /* EXPORT DELETE END */


150 /*
151  * DesN crypt packaged for use as a cipher entry
152  */
153 static OM_uint32
154 __dh_desN_crypt(gss_buffer_t buf, dh_key_set_t keys, cipher_mode_t cipher_mode)
155 {
156         int stat = DESERR_BADPARAM;
161 /* EXPORT DELETE START */
157         int encrypt_flag = (cipher_mode == ENCIPHER);
158         unsigned mode = (encrypt_flag ? DES_ENCRYPT : DES_DECRYPT) | DES_HW;
159         des_block ivec;

161         if (keys->dh_key_set_len < 1)
162                 return (DH_BADARG_FAILURE);

164         /*
165          * We all ways start of with ivec set to zeros. There is no
166          * good way to maintain ivecs since packets could be out of sequence
167          * duplicated or worst of all lost. Under these conditions the
168          * higher level protocol would have to some how resync the ivecs
169          * on both sides and start again. Theres no mechanism for this in
170          * GSS.
171          */
172         memset(&ivec, 0, sizeof (ivec));

174         /* Do the encryption/decryption */
175         stat = __desN_crypt(keys->dh_key_set_val, keys->dh_key_set_len,
176                             (char *)buf->value, buf->length, mode, ivec.c);
182 /* EXPORT DELETE END */

178         if (DES_FAILED(stat))
179                 return (DH_CIPHER_FAILURE);

181         return (DH_SUCCESS);
182 }

184 /*
185  * Package up plain des cbc crypt for use as a cipher entry.
186  */
187 static OM_uint32
188 __dh_des_crypt(gss_buffer_t buf, dh_key_set_t keys, cipher_mode_t cipher_mode)
189 {
190         int stat = DESERR_BADPARAM;
197 /* EXPORT DELETE START */
191         int encrypt_flag = (cipher_mode == ENCIPHER);
192         unsigned mode = (encrypt_flag ? DES_ENCRYPT : DES_DECRYPT) | DES_HW;
193         des_block ivec;

195         if (keys->dh_key_set_len < 1)
196                 return (DH_BADARG_FAILURE);

198         /*  Set the ivec to zeros and then cbc crypt the result */
199         memset(&ivec, 0, sizeof (ivec));
200         stat = cbc_crypt(keys->dh_key_set_val[0].c, (char *)buf->value,
```

```
 201                         buf->length, mode, ivec.c);
 209 /* EXPORT DELETE END */

 203         if (DES_FAILED(stat))
 204                 return (DH_CIPHER_FAILURE);

 206         return (DH_SUCCESS);
 207 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3090 Thu Jul 11 01:29:13 2013
new/usr/src/lib/gss_mechs/mech_dh/backend/mech/dhmech.c
first pass
**********************************************************
```

```
/*
 * 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 2004 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma ident   "%Z%%M% %I%     %E% SMI"

#include "dh_gssapi.h"
#include <stdlib.h>

/*
 * gss_config structure for Diffie-Hellman family of mechanisms.
 * This structure is defined in mechglueP.h and defines the entry points
 * that libgss uses to call a backend.
 */
static struct gss_config dh_mechanism = {
        {0, 0},                         /* OID for mech type. */
        0,
        __dh_gss_acquire_cred,
        __dh_gss_release_cred,
        __dh_gss_init_sec_context,
        __dh_gss_accept_sec_context,
/* EXPORT DELETE START */ /* CRYPT DELETE START */
        __dh_gss_unseal,
/* EXPORT DELETE END */ /* CRYPT DELETE END */
        __dh_gss_process_context_token,
        __dh_gss_delete_sec_context,
        __dh_gss_context_time,
        __dh_gss_display_status,
        NULL, /* Back ends don't implement this */
        __dh_gss_compare_name,
        __dh_gss_display_name,
        __dh_gss_import_name,
        __dh_gss_release_name,
        __dh_gss_inquire_cred,
        NULL, /* Back ends don't implement this */
/* EXPORT DELETE START */ /* CRYPT DELETE START */
        __dh_gss_seal,
/* EXPORT DELETE END */ /* CRYPT DELETE END */
        __dh_gss_export_sec_context,
```



```
        __dh_gss_import_sec_context,
        __dh_gss_inquire_cred_by_mech,
        __dh_gss_inquire_names_for_mech,
        __dh_gss_inquire_context,
        __dh_gss_internal_release_oid,
        __dh_gss_wrap_size_limit,
        __dh_pname_to_uid,
        NULL,  /* __gss_userok */
        __dh_gss_export_name,
/* EXPORT DELETE START */
/* CRYPT DELETE START */
/*
 * This block comment is Sun Proprietary: Need-To-Know.
 * What we are doing is leaving the seal and unseal entry points
 * in an obvious place before sign and unsign for the Domestic customer
 * of the Solaris Source Product. The Domestic customer of the Solaris Source
 * Product will have to deal with the problem of creating exportable libgss
 * binaries.
 * In the binary product that Sun builds, these entry points are elsewhere,
 * and bracketed with special comments so that the CRYPT_SRC and EXPORT_SRC
 * targets delete them.
 */
#if 0
/* CRYPT DELETE END */
        __dh_gss_seal,
        __dh_gss_unseal,
/* CRYPT DELETE START */
#endif /* 0 */
/* CRYPT DELETE END */
/* EXPORT DELETE END */
        __dh_gss_sign,
        __dh_gss_verify,
        NULL, /* gss_store_cred() -- DH lacks this for now */
};
______unchanged_portion_omitted_
```

```
**********************************************************
    1699 Thu Jul 11 01:29:13 2013
new/usr/src/lib/gss_mechs/mech_dummy/Makefile
first pass
**********************************************************
```

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

#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

#
# lib/gss_mechs/mech_dummy/Makefile
#
#
# This make file will build mech_dummy.so.1. This shared object
# contains all the functionality needed to support the Dummy GSS-API
# mechanism.
#

include ../../../Makefile.master

SUBDIRS= $(MACH) $(BUILD64) $(MACH64)

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

HDRS=

CHECKHDRS= $(HDRS:%.h=%.check)

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

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

.KEEP_STATE:

all:    .WAIT $(SUBDIRS)

lint:   .WAIT $(SUBDIRS)
```



```
install: all

install_h:

DUPLICATE_SRC = dmech.c

clean clobber:  $(SUBDIRS)

check: $(CHECKHDRS)

# include library targets
# include ../../Makefile.targ

$(MACH) $(MACH64):      FRC
        @cd $@; pwd; $(MAKE) $(TARGET)

FRC:


# EXPORT DELETE START
# Special target to clean up the source tree for export distribution
# Warning: This target changes the source tree
EXPORT_SRC:
        $(RM) Makefile+ mech/dmech.c+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < mech/dmech.c > mech/dmech.c+
        $(MV) mech/dmech.c+ mech/dmech.c
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile mech/dmech.c

# CRYPT DELETE START
# Special target to clean up the source tree for export distribution
# Warning: This target changes the source tree
CRYPT_SRC:
        $(RM) Makefile+ mech/dmech.c+
        sed -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
                < mech/dmech.c > mech/dmech.c+
        $(MV) mech/dmech.c+ mech/dmech.c
        sed -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile mech/dmech.c

# CRYPT DELETE END
# EXPORT DELETE END
```

```
**********************************************************
   34801 Thu Jul 11 01:29:14 2013
new/usr/src/lib/gss_mechs/mech_dummy/mech/dmech.c
first pass
**********************************************************
```

```
/*
 * 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
 */
#pragma ident   "%Z%%M% %I%     %E% SMI"

/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * A module that implements a dummy security mechanism.
 * It's mainly used to test GSS-API application. Multiple tokens
 * exchanged during security context establishment can be
 * specified through dummy_mech.conf located in /etc.
 *
 */
/* EXPORT DELETE START */ /* CRYPT DELETE START */
#ifndef lint
#define dummy_gss_accept_sec_context \
                dummy_867227349
#define dummy_gss_acquire_cred \
                dummy_352458907
#define dummy_gss_add_cred \
                dummy_911432290
#define dummy_gss_compare_name \
                dummy_396663848
#define dummy_gss_context_time \
                dummy_955669998
#define dummy_gss_delete_sec_context \
                dummy_440868788
#define dummy_gss_display_name \
                dummy_999874939
#define dummy_gss_display_status \
                dummy_485073729
#define dummy_gss_export_sec_context \
                dummy_1044079879
#define dummy_gss_import_name \
                dummy_529311438
#define dummy_gss_import_sec_context \
                dummy_14542996
#define dummy_gss_indicate_mechs \
                dummy_573516378
#define dummy_gss_init_sec_context \
                dummy_58780705
#define dummy_gss_inquire_context \
```



```
                dummy_617721319
#define dummy_gss_inquire_cred \
                dummy_102985645
#define dummy_gss_inquire_cred_by_mech \
                dummy_661926260
#define dummy_gss_inquire_names_for_mech \
                dummy_147190586
#define dummy_gss_internal_release_oid \
                dummy_706163968
#define dummy_gss_process_context_token \
                dummy_191395526
#define dummy_gss_release_cred \
                dummy_750368909
#define dummy_gss_release_name \
                dummy_235600467
#define dummy_gss_seal \
                dummy_794573849
#define dummy_gss_sign \
                dummy_279838176
#define dummy_gss_unseal \
                dummy_838778790
#define dummy_gss_verify \
                dummy_324010348
#define dummy_gss_wrap_size_limit \
                dummy_882983731
#define dummy_pname_to_uid \
                dummy_345475423
#endif
/* EXPORT DELETE END */ /* CRYPT DELETE END */

#include <stdio.h>
#include <stdlib.h>
#include <gssapiP_dummy.h>
#include <mechglueP.h>
#include <gssapi_err_generic.h>

#define dummy_context_name_len  19
/* private routines for dummy_mechanism */
static dummy_token_t make_dummy_token(char *name);
static void free_dummy_token(dummy_token_t *token);
static gss_buffer_desc make_dummy_token_buffer(char *name);
static gss_buffer_desc make_dummy_token_msg(void *data, int datalen);
static int der_length_size(int length);
static void der_write_length(unsigned char ** buf, int length);
static int der_read_length(unsigned char **buf, int *bufsize);
static int g_token_size(gss_OID mech, unsigned int body_size);
static void g_make_token_header(gss_OID mech, int body_size,
                                unsigned char **buf, int tok_type);
static int g_verify_token_header(gss_OID mech, int *body_size,
                                unsigned char **buf_in, int tok_type,
                                int toksize);


/* private global variables */
static char dummy_srcname[] = "dummy source";
static OM_uint32 dummy_flags;
static int token_nums;

/*
 * The Mech OID:
 * { iso(1) org(3) internet(6) dod(1) private(4) enterprises(1) sun(42)
 *  products(2) gssapi(26) mechtypes(1) dummy(2) }
 */
static struct gss_config dummy_mechanism =
        {{10, "\053\006\001\004\001\052\002\032\001\002"},
        NULL,
```

```
126         dummy_gss_acquire_cred,
127         dummy_gss_release_cred,
128         dummy_gss_init_sec_context,
129         dummy_gss_accept_sec_context,
132 /* EXPORT DELETE START */ /* CRYPT DELETE START */
130         dummy_gss_unseal,
134 /* EXPORT DELETE END */ /* CRYPT DELETE END */
131         dummy_gss_process_context_token,
132         dummy_gss_delete_sec_context,
133         dummy_gss_context_time,
134         dummy_gss_display_status,
135         dummy_gss_indicate_mechs,
136         dummy_gss_compare_name,
137         dummy_gss_display_name,
138         dummy_gss_import_name,
139         dummy_gss_release_name,
140         dummy_gss_inquire_cred,
141         dummy_gss_add_cred,
146 /* EXPORT DELETE START */ /* CRYPT DELETE START */
142         dummy_gss_seal,
148 /* EXPORT DELETE END */ /* CRYPT DELETE END */
143         dummy_gss_export_sec_context,
144         dummy_gss_import_sec_context,
145         dummy_gss_inquire_cred_by_mech,
146         dummy_gss_inquire_names_for_mech,
147         dummy_gss_inquire_context,
148         dummy_gss_internal_release_oid,
149         dummy_gss_wrap_size_limit,
150         dummy_pname_to_uid,
151         NULL,   /* __gss_userok */
152         NULL,   /* _export name */
159 /* EXPORT DELETE START */
160 /* CRYPT DELETE START */
161 #if 0
162 /* CRYPT DELETE END */
163         dummy_gss_seal,
164         dummy_gss_unseal,
165 /* CRYPT DELETE START */
166 #endif
167 /* CRYPT DELETE END */
168 /* EXPORT DELETE END */
153         dummy_gss_sign,
154         dummy_gss_verify,
155         NULL,   /* _store_cred */
156 };
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2488 Thu Jul 11 01:29:15 2013
new/usr/src/lib/gss_mechs/mech_krb5/Makefile
first pass
**********************************************************
```

```
#
# 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 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# ident "%Z%%M% %I%     %E% SMI"
#

#
# This make file will build mech_krb5.so.1. This shared object
# contains all the functionality needed to support the Kereros V5 GSS-API
# mechanism. No other Kerberos libraries are needed.
#

include ../../../Makefile.master

SUBDIRS =       $(MACH)
$(BUILD64)SUBDIRS += $(MACH64)

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

GREP= find . \( -name SCCS -prune -o -name '*.[ch]' \) -print | sort | xargs gre

sparcv9_C_PICFLAGS =  -K PIC
TEXT_DOMAIN = SUNW_OST_NETRPC
POFILE = mech_krb5.po
POFILES = generic.po

HDRS=

CHECKHDRS= $(HDRS:%.h=%.check)

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

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



```
.KEEP_STATE:

all clean lint:    $(SUBDIRS)

install: install_dir all .WAIT $(SUBDIRS)

# override ROOTLIBDIR and ROOTLINKS
ROOTLIBDIR=     $(ROOT)/usr/lib/gss

install_dir:    $(ROOTLIBDIR) $(BUILD64)

install_h:

clobber: $(SUBDIRS)
        $(RM) $(POFILE) $(POFILES)

check: $(CHECKHDRS)

do_pkg:
        cd pkg ; pwd ; $(MAKE) install

$(ROOTLIBDIR):
        $(INS.dir)


# include library targets
# include ../../Makefile.targ

$(SUBDIRS):     FRC
        @cd $@; pwd; $(MAKE) $(TARGET)

FRC:

# EXPORT DELETE START
# Special target to clean up the source tree for export distribution
# Warning: This target changes the source tree
EXPORT_SRC:
        $(RM) Makefile+ Makefile.mech_krb5+\
                crypto/des/afsstring2key.c+ \
                crypto/des/string2key.c+ \
                mech/krb5_gss_glue.c+

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < crypto/des/afsstring2key.c > crypto/des/afsstring2key.c+
        $(MV) crypto/des/afsstring2key.c+ crypto/des/afsstring2key.c

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < crypto/des/string2key.c > crypto/des/string2key.c+
        $(MV) crypto/des/string2key.c+ crypto/des/string2key.c

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < mech/krb5_gss_glue.c > mech/krb5_gss_glue.c+
        $(MV) mech/krb5_gss_glue.c+ mech/krb5_gss_glue.c

        $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile.mech_krb5 > Makefile.mech_krb5+
        $(MV) Makefile.mech_krb5+ Makefile.mech_krb5

        $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile

        $(CHMOD) 444 Makefile Makefile.mech_krb5 \
                crypto/des/afsstring2key.c \
                crypto/des/string2key.c \
                mech/krb5_gss_glue.c
```

```
130 # CRYPT DELETE START
131 # Special target to clean up the source tree for domestic distribution
132 # Warning: This target changes the source tree
133 CRYPT_SRC:
134         $(RM) Makefile+ mech/krb5_gss_glue.c+
136         $(SED) -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
137                 > mech/krb5_gss_glue.c+ < mech/krb5_gss_glue.c
138         $(MV) mech/krb5_gss_glue.c+ mech/krb5_gss_glue.c
140         $(SED) -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d"      \
141                         < Makefile                                      \
142             | $(SED) -e "/EXPORT DELETE/d"                              \
143                         > Makefile+
144         $(MV) Makefile+ Makefile
146         $(CHMOD) 444 mech/krb5_gss_glue.c Makefile
148 # CRYPT DELETE END
149 # EXPORT DELETE END
153 FRC:
```

```
 95 _msg: $(MSGDOMAIN) .WAIT $(POFILE)
 96         $(RM) $(MSGDOMAIN)/$(POFILE)
 97         $(CP) $(POFILE) $(MSGDOMAIN)
 99 $(POFILE): $(DERIVED_FILES) .WAIT $(POFILES)
100         $(RM) $@
101         $(CAT) $(POFILES) > $@
103 generic.po: FRC
104         $(RM) messages.po
105         -$(XGETTEXT) $(XGETFLAGS) `$(GREP) -s -l gettext`
106         $(SED) "/^domain/d" messages.po > $@
107         $(RM) messages.po
109 $(MSGDOMAIN):
110         $(INS.dir)
```

****
15659 Thu Jul 11 01:29:15 2013
new/usr/src/lib/gss_mechs/mech_krb5/crypto/des/afsstring2key.c
first pass
****

```
/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */


/*
 * lib/crypto/des/string2key.c
 *
 * based on lib/crypto/des/string2key.c from MIT V5
 * and on lib/des/afs_string_to_key.c from UMD.
 * constructed by Mark Eichin, Cygnus Support, 1995.
 * made thread-safe by Ken Raeburn, MIT, 2001.
 */

/*
 * Copyright 2001 by the Massachusetts Institute of Technology.
 * All Rights Reserved.
 *
 * Export of this software from the United States of America may
 *   require a specific license from the United States Government.
 *   It is the responsibility of any person or organization contemplating
 *   export to obtain such a license before exporting.
 *
 * WITHIN THAT CONSTRAINT, permission to use, copy, modify, and
 * distribute this software and its documentation for any purpose and
 * without fee is hereby granted, provided that the above copyright
 * notice appear in all copies and that both that copyright notice and
 * this permission notice appear in supporting documentation, and that
 * the name of M.I.T. not be used in advertising or publicity pertaining
 * to distribution of the software without specific, written prior
 * permission.  Furthermore if you modify this software you must label
 * your software as modified software and not distribute it in such a
 * fashion that it might be confused with the original M.I.T. software.
 * M.I.T. makes no representations about the suitability of
 * this software for any purpose.  It is provided "as is" without express
 * or implied warranty.
 */

/*
 * Copyright (C) 1998 by the FundsXpress, INC.
 *
 * All rights reserved.
 *
 * Export of this software from the United States of America may require
 * a specific license from the United States Government.  It is the
 * responsibility of any person or organization contemplating export to
 * obtain such a license before exporting.
 *
 * WITHIN THAT CONSTRAINT, permission to use, copy, modify, and
 * distribute this software and its documentation for any purpose and
 * without fee is hereby granted, provided that the above copyright
 * notice appear in all copies and that both that copyright notice and
 * this permission notice appear in supporting documentation, and that
 * the name of FundsXpress. not be used in advertising or publicity pertaining
 * to distribution of the software without specific, written prior
 * permission.  FundsXpress makes no representations about the suitability of
 * this software for any purpose.  It is provided "as is" without express
 * or implied warranty.
 *
 * THIS SOFTWARE IS PROVIDED ''AS IS'' AND WITHOUT ANY EXPRESS OR
```



```
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.
 */

#include "k5-int.h"
#include "des_int.h"
#include <ctype.h>

#define afs_crypt mit_afs_crypt
char *afs_crypt (const char *, const char *, char *);

#undef min
#define min(a,b) ((a)>(b)?(b):(a))

/*ARGSUSED*/
krb5_error_code
mit_afs_string_to_key (krb5_context context,
                       krb5_keyblock *keyblock, const krb5_data *data,
                       const krb5_data *salt)
{
    /* Solaris Kerberos */
    krb5_error_code retval = KRB5_PROG_ETYPE_NOSUPP;
/* EXPORT DELETE START */
  /* totally different approach from MIT string2key. */
  /* much of the work has already been done by the only caller
     which is mit_des_string_to_key; in particular, *keyblock is already
     set up. */

    char *realm = salt->data;
    unsigned int i, j;
    krb5_octet *key = keyblock->contents;
    /* Solaris Kerberos */
    krb5_keyblock usekey;

    if (data->length <= 8) {
      /* One block only.  Run afs_crypt and use the first eight
         returned bytes after the copy of the (fixed) salt.

         Since the returned bytes are alphanumeric, the output is
         limited to 2**48 possibilities; for each byte, only 64
         possible values can be used.  */
      unsigned char password[9]; /* trailing nul for crypt() */
      char afs_crypt_buf[16];

      memset (password, 0, sizeof (password));
      memcpy (password, realm, min (salt->length, 8));
      for (i=0; i<8; i++)
        if (isupper(password[i]))
          password[i] = tolower(password[i]);
      for (i=0; i<data->length; i++)
        password[i] ^= data->data[i];
      for (i=0; i<8; i++)
        if (password[i] == '\0')
          password[i] = 'X';
      password[8] = '\0';
      /* Out-of-bounds salt characters are equivalent to a salt string
         of "p1".  */
      strncpy((char *) key,
              (char *) afs_crypt((char *) password, "#~", afs_crypt_buf) + 2,
              8);
      for (i=0; i<8; i++)
        key[i] <<= 1;
      /* now fix up key parity again */
      mit_des_fixup_key_parity(key);
      /* clean & free the input string */
      memset(password, 0, (size_t) sizeof(password));
```

```
128       /* Solaris Kerberos: Success */
129       retval = 0;
130     } else {
131       /* Multiple blocks.  Do a CBC checksum, twice, and use the
132          result as the new key.  */
133       mit_des_cblock ikey, tkey;
134       unsigned int pw_len = salt->length+data->length;
135       unsigned char *password = malloc(pw_len+1);
136       if (!password) return ENOMEM;

138       /* Some bound checks from the original code are elided here as
139          the malloc above makes sure we have enough storage. */
140       memcpy (password, data->data, data->length);
141       for (i=data->length, j = 0; j < salt->length; i++, j++) {
142         password[i] = realm[j];
143         if (isupper(password[i]))
144           password[i] = tolower(password[i]);
145       }
146
147       memcpy (ikey, "kerberos", sizeof(ikey));
148       memcpy (tkey, ikey, sizeof(tkey));
149       mit_des_fixup_key_parity (tkey);

151     /* Solaris Kerberos */
152       usekey.enctype = ENCTYPE_DES_CBC_CRC;
153       usekey.contents = tkey;
154       usekey.length = 8;
155       retval = mit_des_cbc_cksum (context, (unsigned char *)password,
156                                   tkey, i, &usekey, ikey);

158       memcpy (ikey, tkey, sizeof(ikey));
159       mit_des_fixup_key_parity (tkey);
160       /* Solaris Kerberos */
161       if (usekey.hKey != CK_INVALID_HANDLE) {
162          (void) C_DestroyObject(krb_ctx_hSession(context), usekey.hKey);
163          usekey.hKey = CK_INVALID_HANDLE;
164       }
165       usekey.contents = tkey;
166       usekey.length = 8;
167       retval = mit_des_cbc_cksum (context, (unsigned char *) password,
168                                   key, i, &usekey, ikey);

170       /* now fix up key parity again */
171       mit_des_fixup_key_parity(key);
172
173       /* Solaris Kerberos */
174       if (usekey.hKey != CK_INVALID_HANDLE) {
175          (void) C_DestroyObject(krb_ctx_hSession(context), usekey.hKey);
176          usekey.hKey = CK_INVALID_HANDLE;
177       }
178       /* clean & free the input string */
179       memset(password, 0, (size_t) pw_len);
180       krb5_xfree(password);
181     }
182 #if 0
183     /* must free here because it was copied for this special case */
184     krb5_xfree(salt->data);
185 #endif

188 /* EXPORT DELETE END */
187     return retval;
188 }


191 /* Portions of this code:
```



```
192    Copyright 1989 by the Massachusetts Institute of Technology
193    */
194
195 /*
196  * Copyright (c) 1990 Regents of The University of Michigan.
197  * All Rights Reserved.
198  *
199  * Permission to use, copy, modify, and distribute this software
200  * and its documentation for any purpose and without fee is hereby
201  * granted, provided that the above copyright notice appears in all
202  * copies and that both that copyright notice and this permission
203  * notice appear in supporting documentation, and that the name of
204  * The University of Michigan not be used in advertising or
205  * publicity pertaining to distribution of the software without
206  * specific, written prior permission. This software is supplied as
207  * is without expressed or implied warranties of any kind.
208  *
209  *      ITD Research Systems
210  *      University of Michigan
211  *      535 W. William Street
212  *      Ann Arbor, Michigan
213  *      +1-313-936-2652
214  *      netatalk@terminator.cc.umich.edu
215  */

219 /* EXPORT DELETE START */

217 static void krb5_afs_crypt_setkey (char*, char*, char(*)[48]);
218 static void krb5_afs_encrypt (char*,char*,char (*)[48]);

220 /*
221  * Initial permutation,
222  */
223 static const char       IP[] = {
224         58,50,42,34,26,18,10, 2,
225         60,52,44,36,28,20,12, 4,
226         62,54,46,38,30,22,14, 6,
227         64,56,48,40,32,24,16, 8,
228         57,49,41,33,25,17, 9, 1,
229         59,51,43,35,27,19,11, 3,
230         61,53,45,37,29,21,13, 5,
231         63,55,47,39,31,23,15, 7,
232 };
_____unchanged_portion_omitted_
605 /* EXPORT DELETE END */
```

```
**********************************************************
    5154 Thu Jul 11 01:29:16 2013
new/usr/src/lib/gss_mechs/mech_krb5/crypto/des/string2key.c
first pass
**********************************************************
```

```
   1 /*
   2  * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
   3  * Use is subject to license terms.
   4  */

   6 #pragma ident   "%Z%%M% %I%     %E% SMI"

   8 /*
   9  * lib/crypto/des/string2key.c
  10  *
  11  * Copyright 1990,1991 by the Massachusetts Institute of Technology.
  12  * All Rights Reserved.
  13  *
  14  * Export of this software from the United States of America may
  15  *   require a specific license from the United States Government.
  16  *   It is the responsibility of any person or organization contemplating
  17  *   export to obtain such a license before exporting.
  18  *
  19  * WITHIN THAT CONSTRAINT, permission to use, copy, modify, and
  20  * distribute this software and its documentation for any purpose and
  21  * without fee is hereby granted, provided that the above copyright
  22  * notice appear in all copies and that both that copyright notice and
  23  * this permission notice appear in supporting documentation, and that
  24  * the name of M.I.T. not be used in advertising or publicity pertaining
  25  * to distribution of the software without specific, written prior
  26  * permission.  M.I.T. makes no representations about the suitability of
  27  * this software for any purpose.  It is provided "as is" without express
  28  * or implied warranty.
  29  */

  31 #include <k5-int.h>
  32 #include <des_int.h>

  34 /*
  35         converts the string pointed to by "data" into an encryption key
  36         of type "enctype".  *keyblock is filled in with the key info;
  37         in particular, keyblock->contents is to be set to allocated storage.
  38         It is the responsibility of the caller to release this storage
  39         when the generated key no longer needed.

  41         The routine may use "salt" to seed or alter the conversion
  42         algorithm.

  44         If the particular function called does not know how to make a
  45         key of type "enctype", an error may be returned.

  47         returns: errors
  48  */

  50 krb5_error_code
  51 mit_des_string_to_key_int (krb5_context context,
  52         krb5_keyblock *keyblock,
  53         const krb5_data *data,
  54         const krb5_data *salt)
  55 {
  56     krb5_error_code retval = KRB5_PROG_ETYPE_NOSUPP;
  57 /* EXPORT DELETE START */
  57     register char *str, *copystr;
  58     register krb5_octet *key;
  59     register unsigned temp;
  60     register long i;
```



```
  61     register int j;
  62     register long length;
  63     unsigned char *k_p;
  64     int forward;
  65     register char *p_char;
  66     char k_char[64];

  68 #ifndef min
  69 #define min(A, B) ((A) < (B) ? (A): (B))
  70 #endif

  72     keyblock->magic = KV5M_KEYBLOCK;
  73     keyblock->length = sizeof(mit_des_cblock);
  74     key = keyblock->contents;

  76     if (salt
  77         && (salt->length == SALT_TYPE_AFS_LENGTH
  78             /* XXX  Yuck!  Aren't we done with this yet?  */
  79             || salt->length == (unsigned) -1)) {
  80         krb5_data afssalt;
  81         char *at;

  83         afssalt.data = salt->data;
  84         at = strchr(afssalt.data, '@');
  85         if (at) {
  86             *at = 0;
  87             afssalt.length = at - afssalt.data;
  88         } else
  89             afssalt.length = strlen(afssalt.data);
  90         return mit_afs_string_to_key(context, keyblock, data, &afssalt);
  91     }

  93     length = data->length + (salt ? salt->length : 0);

  95     copystr = malloc((size_t) length);
  96     if (!copystr) {
  97         return ENOMEM;
  98     }

 100     (void) memcpy(copystr, (char *) data->data, data->length);
 101     if (salt)
 102         (void) memcpy(copystr + data->length, (char *)salt->data, salt->length);

 104     /* convert to des key */
 105     forward = 1;
 106     p_char = k_char;

 108     /* init key array for bits */
 109     (void) memset(k_char,0,sizeof(k_char));

 111 #if 0
 112     if (mit_des_debug)
 113         fprintf(stdout,
 114                 "\n\ninput str length = %d  string = %*s\nstring = 0x ",
 115                 length,length,str);
 116 #endif

 118     str = copystr;

 120     /* get next 8 bytes, strip parity, xor */
 121     for (i = 1; i <= length; i++) {
 122         /* get next input key byte */
 123         temp = (unsigned int) *str++;
 124 #if 0
 125         if (mit_des_debug)
 126             fprintf(stdout,"%02x ",temp & 0xff);
```

```
#endif
        /* loop through bits within byte, ignore parity */
        for (j = 0; j <= 6; j++) {
            if (forward)
                *p_char++ ^= (int) temp & 01;
            else
                *--p_char ^= (int) temp & 01;
            temp = temp >> 1;
        }

        /* check and flip direction */
        if ((i%8) == 0)
            forward = !forward;
    }

    /* now stuff into the key mit_des_cblock, and force odd parity */
    p_char = k_char;
    k_p = (unsigned char *) key;

    for (i = 0; i <= 7; i++) {
        temp = 0;
        for (j = 0; j <= 6; j++)
            temp |= *p_char++ << (1+j);
        *k_p++ = (unsigned char) temp;
    }

    /* fix key parity */
    mit_des_fixup_key_parity(key);
    if (mit_des_is_weak_key(key))
        ((krb5_octet *)key)[7] ^= 0xf0;

    retval = mit_des_cbc_cksum(context, (unsigned char*)copystr, key,
        length, keyblock, key);

    /* clean & free the input string */
    (void) memset(copystr, 0, (size_t) length);
    krb5_xfree(copystr);

    /* now fix up key parity again */
    mit_des_fixup_key_parity(key);
    if (mit_des_is_weak_key(key))
        ((krb5_octet *)key)[7] ^= 0xf0;

    /*
     * Because this routine actually modifies the original keyblock
     * in place we cannot use the PKCS#11 key object handle created earlier.
     * Destroy the existing object handle associated with the key,
     * a correct handle will get created when the key is actually
     * used for the first time.
     */
     if (keyblock->hKey != CK_INVALID_HANDLE) {
        (void)C_DestroyObject(krb_ctx_hSession(context), keyblock->hKey);
        keyblock->hKey = CK_INVALID_HANDLE;
     }

/* EXPORT DELETE END */
    return retval;
}
```

*unchanged_portion_omitted_*

```
**********************************************************
   40149 Thu Jul 11 01:29:16 2013
new/usr/src/lib/gss_mechs/mech_krb5/mech/krb5_gss_glue.c
first pass
**********************************************************
```

```
/*
 * Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
 */
/*
 * Copyright 1993 by OpenVision Technologies, Inc.
 *
 * Permission to use, copy, modify, distribute, and sell this software
 * and its documentation for any purpose is hereby granted without fee,
 * provided that the above copyright notice appears in all copies and
 * that both that copyright notice and this permission notice appear in
 * supporting documentation, and that the name of OpenVision not be used
 * in advertising or publicity pertaining to distribution of the software
 * without specific, written prior permission. OpenVision makes no
 * representations about the suitability of this software for any
 * purpose.  It is provided "as is" without express or implied warranty.
 *
 * OPENVISION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
 * INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO
 * EVENT SHALL OPENVISION BE LIABLE FOR ANY SPECIAL, INDIRECT OR
 * CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF
 * USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
 * OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
 * PERFORMANCE OF THIS SOFTWARE.
 */

/*
 * $Id: krb5_gss_glue.c 18262 2006-06-29 04:38:48Z tlyu $
 */

#include "gssapiP_krb5.h"
#include "mglueP.h"
#include <syslog.h>

/** mechglue wrappers **/

static OM_uint32 k5glue_acquire_cred
(void *, OM_uint32*,        /* minor_status */
            gss_name_t,       /* desired_name */
            OM_uint32,        /* time_req */
            gss_OID_set,      /* desired_mechs */
            gss_cred_usage_t, /* cred_usage */
            gss_cred_id_t*,   /* output_cred_handle */
            gss_OID_set*,     /* actual_mechs */
            OM_uint32*        /* time_rec */
           );

static OM_uint32 k5glue_release_cred
(void *, OM_uint32*,        /* minor_status */
            gss_cred_id_t*    /* cred_handle */
           );

static OM_uint32 k5glue_init_sec_context
(void *, OM_uint32*,        /* minor_status */
            gss_cred_id_t,    /* claimant_cred_handle */
            gss_ctx_id_t*,    /* context_handle */
            gss_name_t,       /* target_name */
            gss_OID,          /* mech_type */
            OM_uint32,        /* req_flags */
            OM_uint32,        /* time_req */
            gss_channel_bindings_t,
                              /* input_chan_bindings */
```



```
            gss_buffer_t,     /* input_token */
            gss_OID*,         /* actual_mech_type */
            gss_buffer_t,     /* output_token */
            OM_uint32*,       /* ret_flags */
            OM_uint32*        /* time_rec */
           );

static OM_uint32 k5glue_accept_sec_context
(void *, OM_uint32*,        /* minor_status */
            gss_ctx_id_t*,    /* context_handle */
            gss_cred_id_t,    /* verifier_cred_handle */
            gss_buffer_t,     /* input_token_buffer */
            gss_channel_bindings_t,
                              /* input_chan_bindings */
            gss_name_t*,      /* src_name */
            gss_OID*,         /* mech_type */
            gss_buffer_t,     /* output_token */
            OM_uint32*,       /* ret_flags */
            OM_uint32*,       /* time_rec */
            gss_cred_id_t*    /* delegated_cred_handle */
           );

static OM_uint32 k5glue_process_context_token
(void *, OM_uint32*,        /* minor_status */
            gss_ctx_id_t,     /* context_handle */
            gss_buffer_t      /* token_buffer */
           );

static OM_uint32 k5glue_delete_sec_context
(void *, OM_uint32*,        /* minor_status */
            gss_ctx_id_t*,    /* context_handle */
            gss_buffer_t      /* output_token */
           );

static OM_uint32 k5glue_context_time
(void *, OM_uint32*,        /* minor_status */
            gss_ctx_id_t,     /* context_handle */
            OM_uint32*        /* time_rec */
           );

static OM_uint32 k5glue_sign
(void *, OM_uint32*,        /* minor_status */
            gss_ctx_id_t,     /* context_handle */
            int,              /* qop_req */
            gss_buffer_t,     /* message_buffer */
            gss_buffer_t      /* message_token */
           );

static OM_uint32 k5glue_verify
(void *, OM_uint32*,        /* minor_status */
            gss_ctx_id_t,     /* context_handle */
            gss_buffer_t,     /* message_buffer */
            gss_buffer_t,     /* token_buffer */
            int*              /* qop_state */
           );

/* EXPORT DELETE START */
static OM_uint32 k5glue_seal
(void *, OM_uint32*,        /* minor_status */
            gss_ctx_id_t,     /* context_handle */
            int,              /* conf_req_flag */
            int,              /* qop_req */
            gss_buffer_t,     /* input_message_buffer */
            int*,             /* conf_state */
            gss_buffer_t      /* output_message_buffer */
           );
```

```
128 static OM_uint32 k5glue_unseal
129 (void *, OM_uint32*,       /* minor_status */
130             gss_ctx_id_t,     /* context_handle */
131             gss_buffer_t,     /* input_message_buffer */
132             gss_buffer_t,     /* output_message_buffer */
133             int*,             /* conf_state */
134             int*              /* qop_state */
135            );
137 /* EXPORT DELETE END */

137 static OM_uint32 k5glue_display_status
138 (void *, OM_uint32*,       /* minor_status */
139             OM_uint32,        /* status_value */
140             int,              /* status_type */
141             gss_OID,          /* mech_type */
142             OM_uint32*,       /* message_context */
143             gss_buffer_t      /* status_string */
144            );

146 static OM_uint32 k5glue_indicate_mechs
147 (void *, OM_uint32*,       /* minor_status */
148             gss_OID_set*      /* mech_set */
149            );

151 static OM_uint32 k5glue_compare_name
152 (void *, OM_uint32*,       /* minor_status */
153             gss_name_t,       /* name1 */
154             gss_name_t,       /* name2 */
155             int*              /* name_equal */
156            );

158 static OM_uint32 k5glue_display_name
159 (void *, OM_uint32*,      /* minor_status */
160             gss_name_t,      /* input_name */
161             gss_buffer_t,    /* output_name_buffer */
162             gss_OID*         /* output_name_type */
163            );

165 static OM_uint32 k5glue_import_name
166 (void *, OM_uint32*,       /* minor_status */
167             gss_buffer_t,     /* input_name_buffer */
168             gss_OID,          /* input_name_type */
169             gss_name_t*       /* output_name */
170            );

172 static OM_uint32 k5glue_release_name
173 (void *, OM_uint32*,       /* minor_status */
174             gss_name_t*       /* input_name */
175            );

177 static OM_uint32 k5glue_inquire_cred
178 (void *, OM_uint32 *,      /* minor_status */
179             gss_cred_id_t,    /* cred_handle */
180             gss_name_t *,     /* name */
181             OM_uint32 *,      /* lifetime */
182             gss_cred_usage_t*,/* cred_usage */
183             gss_OID_set *     /* mechanisms */
184            );

186 static OM_uint32 k5glue_inquire_context
187 (void *, OM_uint32*,       /* minor_status */
188             gss_ctx_id_t,     /* context_handle */
189             gss_name_t*,      /* initiator_name */
190             gss_name_t*,      /* acceptor_name */
191             OM_uint32*,       /* lifetime_rec */
```



```
192             gss_OID*,         /* mech_type */
193             OM_uint32*,       /* ret_flags */
194             int*,             /* locally_initiated */
195             int*              /* open */
196            );

198 #if 0
199 /* New V2 entry points */
200 static OM_uint32 k5glue_get_mic
201 (void *, OM_uint32 *,           /* minor_status */
202             gss_ctx_id_t,             /* context_handle */
203             gss_qop_t,                /* qop_req */
204             gss_buffer_t,             /* message_buffer */
205             gss_buffer_t              /* message_token */
206            );

208 static OM_uint32 k5glue_verify_mic
209 (void *, OM_uint32 *,           /* minor_status */
210             gss_ctx_id_t,             /* context_handle */
211             gss_buffer_t,             /* message_buffer */
212             gss_buffer_t,             /* message_token */
213             gss_qop_t *               /* qop_state */
214            );

216 static OM_uint32 k5glue_wrap
217 (void *, OM_uint32 *,           /* minor_status */
218             gss_ctx_id_t,             /* context_handle */
219             int,                      /* conf_req_flag */
220             gss_qop_t,                /* qop_req */
221             gss_buffer_t,             /* input_message_buffer */
222             int *,                    /* conf_state */
223             gss_buffer_t              /* output_message_buffer */
224            );

226 static OM_uint32 k5glue_unwrap
227 (void *, OM_uint32 *,           /* minor_status */
228             gss_ctx_id_t,             /* context_handle */
229             gss_buffer_t,             /* input_message_buffer */
230             gss_buffer_t,             /* output_message_buffer */
231             int *,                    /* conf_state */
232             gss_qop_t *               /* qop_state */
233            );
234 #endif

236 static OM_uint32 k5glue_wrap_size_limit
237 (void *, OM_uint32 *,           /* minor_status */
238             gss_ctx_id_t,             /* context_handle */
239             int,                      /* conf_req_flag */
240             gss_qop_t,                /* qop_req */
241             OM_uint32,                /* req_output_size */
242             OM_uint32 *               /* max_input_size */
243            );

245 #if 0
246 static OM_uint32 k5glue_import_name_object
247 (void *, OM_uint32 *,           /* minor_status */
248             void *,                   /* input_name */
249             gss_OID,                  /* input_name_type */
250             gss_name_t *              /* output_name */
251            );

253 static OM_uint32 k5glue_export_name_object
254 (void *, OM_uint32 *,           /* minor_status */
255             gss_name_t,               /* input_name */
256             gss_OID,                  /* desired_name_type */
257             void * *                  /* output_name */
```

```
258           );
259 #endif

261 static OM_uint32 k5glue_add_cred
262 (void *, OM_uint32 *,           /* minor_status */
263             gss_cred_id_t,              /* input_cred_handle */
264             gss_name_t,                 /* desired_name */
265             gss_OID,                    /* desired_mech */
266             gss_cred_usage_t,           /* cred_usage */
267             OM_uint32,                  /* initiator_time_req */
268             OM_uint32,                  /* acceptor_time_req */
269             gss_cred_id_t *,            /* output_cred_handle */
270             gss_OID_set *,              /* actual_mechs */
271             OM_uint32 *,                /* initiator_time_rec */
272             OM_uint32 *                 /* acceptor_time_rec */
273           );

275 static OM_uint32 k5glue_inquire_cred_by_mech
276 (void *, OM_uint32  *,          /* minor_status */
277             gss_cred_id_t,              /* cred_handle */
278             gss_OID,                    /* mech_type */
279             gss_name_t *,               /* name */
280             OM_uint32 *,                /* initiator_lifetime */
281             OM_uint32 *,                /* acceptor_lifetime */
282             gss_cred_usage_t *          /* cred_usage */
283           );

285 static OM_uint32 k5glue_export_sec_context
286 (void *, OM_uint32 *,           /* minor_status */
287             gss_ctx_id_t *,             /* context_handle */
288             gss_buffer_t                /* interprocess_token */
289             );

291 static OM_uint32 k5glue_import_sec_context
292 (void *, OM_uint32 *,           /* minor_status */
293             gss_buffer_t,               /* interprocess_token */
294             gss_ctx_id_t *              /* context_handle */
295             );

297 krb5_error_code k5glue_ser_init(krb5_context);

299 static OM_uint32 k5glue_internal_release_oid
300 (void *, OM_uint32 *,           /* minor_status */
301             gss_OID *                   /* oid */
302           );

304 static OM_uint32 k5glue_inquire_names_for_mech
305 (void *, OM_uint32 *,           /* minor_status */
306             gss_OID,                    /* mechanism */
307             gss_OID_set *               /* name_types */
308           );

310 #if 0
311 static OM_uint32 k5glue_canonicalize_name
312 (void *, OM_uint32  *,          /* minor_status */
313             const gss_name_t,           /* input_name */
314             const gss_OID,              /* mech_type */
315             gss_name_t *                /* output_name */
316         );
317 #endif

319 static OM_uint32 k5glue_export_name
320 (void *, OM_uint32  *,          /* minor_status */
321             const gss_name_t,           /* input_name */
322             gss_buffer_t                /* exported_name */
323         );
```



```
325 /* SUNW15resync - Solaris specific */
326 static OM_uint32 k5glue_store_cred (
327             void *,
328             OM_uint32 *,            /* minor_status */
329             const gss_cred_id_t,    /* input_cred */
330             gss_cred_usage_t,       /* cred_usage */
331             const gss_OID,          /* desired_mech */
332             OM_uint32,              /* overwrite_cred */
333             OM_uint32,              /* default_cred */
334             gss_OID_set *,          /* elements_stored */
335             gss_cred_usage_t *      /* cred_usage_stored */
336           );

338 /* SUNW17PACresync - this decl not needed in MIT but is for Sol */
339 /* Note code is in gsspi_krb5.c */
340 OM_uint32 krb5_gss_inquire_sec_context_by_oid(
341         OM_uint32 *,
342         const gss_ctx_id_t,
343         const gss_OID,
344         gss_buffer_set_t *);

346 static OM_uint32
347 k5glue_userok(
348                     void *,             /* context */
349                     OM_uint32 *,        /* minor_status */
350                     const gss_name_t,   /* pname */
351                     const char *,       /* local user */
352                     int *               /* user ok? */
353         /* */);

355 static OM_uint32
356 k5glue_pname_to_uid(
357                     void *,             /* context */
358                     OM_uint32 *,        /* minor_status */
359                     const gss_name_t,   /* pname */
360                     uid_t *             /* uid */
361         /* */);



366 #if 0
367 static OM_uint32 k5glue_duplicate_name
368 (void *, OM_uint32  *,          /* minor_status */
369             const gss_name_t,           /* input_name */
370             gss_name_t *                /* dest_name */
371          );
372 #endif

374 #if 0
375 static OM_uint32 k5glue_validate_cred
376 (void *, OM_uint32 *,           /* minor_status */
377             gss_cred_id_t               /* cred */
378          );
379 #endif

381 #if 0
382 /*
383  * SUNW15resync
384  * Solaris can't use the KRB5_GSS_CONFIG_INIT macro because of the src
385  * slicing&dicing needs of the "nightly -SD" build.  When it goes away,
386  * we should use it assuming MIT still uses it then.
387  */

389 /*
```

```
 390  * The krb5 mechanism provides two mech OIDs; use this initializer to
 391  * ensure that both dispatch tables contain identical function
 392  * pointers.
 393  */
 394 #define KRB5_GSS_CONFIG_INIT                            \
 395     NULL,                                               \
 396     ...
 397 #endif


 400 static struct gss_config krb5_mechanism = {
 401 #if 0 /* Solaris Kerberos */
 402     100, "kerberos_v5",
 403 #endif
 404     { GSS_MECH_KRB5_OID_LENGTH, GSS_MECH_KRB5_OID },
 405     NULL,
 406     k5glue_acquire_cred,
 407     k5glue_release_cred,
 408     k5glue_init_sec_context,
 409     k5glue_accept_sec_context,
 412 /* EXPORT DELETE START */ /* CRYPT DELETE START */
 410     k5glue_unseal,
 414 /* EXPORT DELETE END */ /* CRYPT DELETE END */
 411     k5glue_process_context_token,
 412     k5glue_delete_sec_context,
 413     k5glue_context_time,
 414     k5glue_display_status,
 415     k5glue_indicate_mechs,
 416     k5glue_compare_name,
 417     k5glue_display_name,
 418     k5glue_import_name,
 419     k5glue_release_name,
 420     k5glue_inquire_cred,
 421     k5glue_add_cred,
 426 /* EXPORT DELETE START */ /* CRYPT DELETE START */
 422     k5glue_seal,
 428 /* EXPORT DELETE END */ /* CRYPT DELETE END */
 423     k5glue_export_sec_context,
 424     k5glue_import_sec_context,
 425     k5glue_inquire_cred_by_mech,
 426     k5glue_inquire_names_for_mech,
 427     k5glue_inquire_context,
 428     k5glue_internal_release_oid,
 429     k5glue_wrap_size_limit,
 430     k5glue_pname_to_uid,
 431     k5glue_userok,
 432     k5glue_export_name,
 439 /* EXPORT DELETE START */
 440 /* CRYPT DELETE START */
 441 #if 0
 442 /* CRYPT DELETE END */
 443     k5glue_seal,
 444     k5glue_unseal,
 445 /* CRYPT DELETE START */
 446 #endif
 447 /* CRYPT DELETE END */
 448 /* EXPORT DELETE END */
 433     k5glue_sign,
 434     k5glue_verify,
 435     k5glue_store_cred,
 436     krb5_gss_inquire_sec_context_by_oid
 437 };

 439 static struct gss_config krb5_mechanism_old = {
 440 #if 0 /* Solaris Kerberos */
 441     200, "kerberos_v5 (pre-RFC OID)",
```



```
 442 #endif
 443     { GSS_MECH_KRB5_OLD_OID_LENGTH, GSS_MECH_KRB5_OLD_OID },
 444     NULL,
 445     k5glue_acquire_cred,
 446     k5glue_release_cred,
 447     k5glue_init_sec_context,
 448     k5glue_accept_sec_context,
 465 /* EXPORT DELETE START */ /* CRYPT DELETE START */
 449     k5glue_unseal,
 467 /* EXPORT DELETE END */ /* CRYPT DELETE END */
 450     k5glue_process_context_token,
 451     k5glue_delete_sec_context,
 452     k5glue_context_time,
 453     k5glue_display_status,
 454     k5glue_indicate_mechs,
 455     k5glue_compare_name,
 456     k5glue_display_name,
 457     k5glue_import_name,
 458     k5glue_release_name,
 459     k5glue_inquire_cred,
 460     k5glue_add_cred,
 479 /* EXPORT DELETE START */ /* CRYPT DELETE START */
 461     k5glue_seal,
 481 /* EXPORT DELETE END */ /* CRYPT DELETE END */
 462     k5glue_export_sec_context,
 463     k5glue_import_sec_context,
 464     k5glue_inquire_cred_by_mech,
 465     k5glue_inquire_names_for_mech,
 466     k5glue_inquire_context,
 467     k5glue_internal_release_oid,
 468     k5glue_wrap_size_limit,
 469     k5glue_pname_to_uid,
 470     k5glue_userok,
 471     k5glue_export_name,
 492 /* EXPORT DELETE START */
 493 /* CRYPT DELETE START */
 494 #if 0
 495 /* CRYPT DELETE END */
 496     k5glue_seal,
 497     k5glue_unseal,
 498 /* CRYPT DELETE START */
 499 #endif
 500 /* CRYPT DELETE END */
 501 /* EXPORT DELETE END */
 472     k5glue_sign,
 473     k5glue_verify,
 474     k5glue_store_cred,
 475     krb5_gss_inquire_sec_context_by_oid
 476 };

 478 static struct gss_config krb5_mechanism_wrong = {
 479 #if 0 /* Solaris Kerberos */
 480     300, "kerberos_v5 (wrong OID)",
 481 #endif
 482     { GSS_MECH_KRB5_WRONG_OID_LENGTH, GSS_MECH_KRB5_WRONG_OID },
 483     NULL,
 484     k5glue_acquire_cred,
 485     k5glue_release_cred,
 486     k5glue_init_sec_context,
 487     k5glue_accept_sec_context,
 518 /* EXPORT DELETE START */ /* CRYPT DELETE START */
 488     k5glue_unseal,
 520 /* EXPORT DELETE END */ /* CRYPT DELETE END */
 489     k5glue_process_context_token,
 490     k5glue_delete_sec_context,
 491     k5glue_context_time,
```

```
 492     k5glue_display_status,
 493     k5glue_indicate_mechs,
 494     k5glue_compare_name,
 495     k5glue_display_name,
 496     k5glue_import_name,
 497     k5glue_release_name,
 498     k5glue_inquire_cred,
 499     k5glue_add_cred,
 532 /* EXPORT DELETE START */ /* CRYPT DELETE START */
 500     k5glue_seal,
 534 /* EXPORT DELETE END */ /* CRYPT DELETE END */
 501     k5glue_export_sec_context,
 502     k5glue_import_sec_context,
 503     k5glue_inquire_cred_by_mech,
 504     k5glue_inquire_names_for_mech,
 505     k5glue_inquire_context,
 506     k5glue_internal_release_oid,
 507     k5glue_wrap_size_limit,
 508     k5glue_pname_to_uid,
 509     k5glue_userok,
 510     k5glue_export_name,
 545 /* EXPORT DELETE START */
 546 /* CRYPT DELETE START */
 547 #if 0
 548 /* CRYPT DELETE END */
 549     k5glue_seal,
 550     k5glue_unseal,
 551 /* CRYPT DELETE START */
 552 #endif
 553 /* CRYPT DELETE END */
 554 /* EXPORT DELETE END */
 511     k5glue_sign,
 512     k5glue_verify,
 513     k5glue_store_cred,
 514     krb5_gss_inquire_sec_context_by_oid
 515 };
______unchanged_portion_omitted_
 950 #endif

 996 /* EXPORT DELETE START */
 952 /* V1 only */
 953 static OM_uint32
 954 k5glue_seal(ctx, minor_status, context_handle, conf_req_flag, qop_req,
 955           input_message_buffer, conf_state, output_message_buffer)
 956     void *ctx;
 957     OM_uint32 *minor_status;
 958     gss_ctx_id_t context_handle;
 959     int conf_req_flag;
 960     int qop_req;
 961     gss_buffer_t input_message_buffer;
 962     int *conf_state;
 963     gss_buffer_t output_message_buffer;
 964 {
 965    return(krb5_gss_seal(minor_status, context_handle,
 966                         conf_req_flag, qop_req, input_message_buffer,
 967                         conf_state, output_message_buffer));
 968 }
1014 /* EXPORT DELETE END */

 970 static OM_uint32
 971 k5glue_sign(ctx, minor_status, context_handle,
 972             qop_req, message_buffer,
 973             message_token)
 974     void *ctx;
 975     OM_uint32 *minor_status;
 976     gss_ctx_id_t context_handle;
```



```
 977     int qop_req;
 978     gss_buffer_t message_buffer;
 979     gss_buffer_t message_token;
 980 {
 981    return(krb5_gss_sign(minor_status, context_handle,
 982                         qop_req, message_buffer, message_token));
 983 }
______unchanged_portion_omitted_
1042 #endif

1090 /* EXPORT DELETE START */
1044 /* V1 only */
1045 static OM_uint32
1046 k5glue_unseal(ctx, minor_status, context_handle, input_message_buffer,
1047           output_message_buffer, conf_state, qop_state)
1048     void *ctx;
1049     OM_uint32 *minor_status;
1050     gss_ctx_id_t context_handle;
1051     gss_buffer_t input_message_buffer;
1052     gss_buffer_t output_message_buffer;
1053     int *conf_state;
1054     int *qop_state;
1055 {
1056    return(krb5_gss_unseal(minor_status, context_handle,
1057                           input_message_buffer, output_message_buffer,
1058                           conf_state, qop_state));
1059 }
1107 /* EXPORT DELETE END */

1061 #if 0
1062 /* V2 */
1063 static OM_uint32
1064 k5glue_unwrap(ctx, minor_status, context_handle, input_message_buffer,
1065           output_message_buffer, conf_state, qop_state)
1066     void *ctx;
1067     OM_uint32           *minor_status;
1068     gss_ctx_id_t        context_handle;
1069     gss_buffer_t        input_message_buffer;
1070     gss_buffer_t        output_message_buffer;
1071     int                 *conf_state;
1072     gss_qop_t           *qop_state;
1073 {
1074     return(krb5_gss_unwrap(minor_status, context_handle, input_message_buffer,
1075                            output_message_buffer, conf_state, qop_state));
1076 }
______unchanged_portion_omitted_
```

```
**********************************************************
    1443 Thu Jul 11 01:29:17 2013
new/usr/src/lib/gss_mechs/mech_spnego/Makefile
first pass
**********************************************************
```

```
#
# 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 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# ident "%Z%%M% %I%     %E% SMI"
#

#
# This make file will build mech_spnego.so.1. This shared object
# contains all the functionality needed to support the
# SPNEGO GSS-API mechanism.
#

include ../../Makefile.lib

SUBDIRS= $(MACH)
$(BUILD64)SUBDIRS += $(MACH64)

HDRS    =       gssapiP_spnego.h
HDRDIR  =       mech

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

.KEEP_STATE:

all clean clobber install lint:  $(SUBDIRS)

$(SUBDIRS): FRC
        @cd $@; pwd; $(MAKE) $(TARGET)

FRC:

# EXPORT DELETE START
# Special target to clean up the source tree for export distribution
# Warning: This target changes the source tree
EXPORT_SRC:
        $(RM) Makefile+ mech/spnego_mech.c+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < mech/spnego_mech.c > mech/spnego_mech.c+
        $(MV) mech/spnego_mech.c+ mech/spnego_mech.c
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile mech/spnego_mech.c

# CRYPT DELETE START
# Special target to clean up the source tree for domestic distribution
# Warning: This target changes the source tree
CRYPT_SRC:
        $(RM) Makefile+ mech/spnego_mech.c+
        sed -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
                < mech/spnego_mech.c > mech/spnego_mech.c+
        $(MV) mech/spnego_mech.c+ mech/spnego_mech.c
        sed -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile mech/spnego_mech.c

# CRYPT DELETE END
# EXPORT DELETE END
```

```
**********************************************************
  101023 Thu Jul 11 01:29:18 2013
new/usr/src/lib/gss_mechs/mech_spnego/mech/spnego_mech.c
first pass
**********************************************************
______unchanged_portion_omitted_
 213 const gss_OID_set_desc * const gss_mech_set_spnego = spnego_oidsets+0;

 215 static int make_NegHints(OM_uint32 *, gss_cred_id_t, gss_buffer_t *);
 216 static int put_neg_hints(unsigned char **, gss_buffer_t, unsigned int);
 217 static OM_uint32
 218 acc_ctx_hints(OM_uint32 *, gss_ctx_id_t *, gss_cred_id_t,
 219               gss_buffer_t *, OM_uint32 *, send_token_flag *);

 221 #ifdef _GSS_STATIC_LINK
 222 int gss_spnegoint_lib_init(void);
 223 void gss_spnegoint_lib_fini(void);
 224 #else
 225 gss_mechanism gss_mech_initialize(void);
 226 #endif /* _GSS_STATIC_LINK */

 228 /*
 229  * The Mech OID for SPNEGO:
 230  * { iso(1) org(3) dod(6) internet(1) security(5)
 231  *  mechanism(5) spnego(2) }
 232  */
 233 static struct gss_config spnego_mechanism =
 234 {
 235         {SPNEGO_OID_LENGTH, SPNEGO_OID},
 236         NULL,
 237         glue_spnego_gss_acquire_cred,
 238         glue_spnego_gss_release_cred,
 239         glue_spnego_gss_init_sec_context,
 240 #ifndef LEAN_CLIENT
 241         glue_spnego_gss_accept_sec_context,
 242 #else
 243         NULL,
 244 #endif  /* LEAN_CLIENT */
 245 /* EXPORT DELETE START */ /* CRYPT DELETE START */
 245         NULL,  /* unseal */
 247 /* EXPORT DELETE END */ /* CRYPT DELETE END */
 246         NULL,                           /* gss_process_context_token */
 247         glue_spnego_gss_delete_sec_context,     /* gss_delete_sec_context */
 248         glue_spnego_gss_context_time,
 249         glue_spnego_gss_display_status,
 250         NULL,                           /* gss_indicate_mechs */
 251         glue_spnego_gss_compare_name,
 252         glue_spnego_gss_display_name,
 253         glue_spnego_gss_import_name, /* glue */
 254         glue_spnego_gss_release_name,
 255         NULL,                           /* gss_inquire_cred */
 256         NULL,                           /* gss_add_cred */
 259 /* EXPORT DELETE START */ /* CRYPT DELETE START */
 257         NULL, /* seal */
 261 /* EXPORT DELETE END */ /* CRYPT DELETE END */
 258 #ifndef LEAN_CLIENT
 259         glue_spnego_gss_export_sec_context,     /* gss_export_sec_context */
 260         glue_spnego_gss_import_sec_context,     /* gss_import_sec_context */
 261 #else
 262         NULL,                           /* gss_export_sec_context */
 263         NULL,                           /* gss_import_sec_context */
 264 #endif /* LEAN_CLIENT */
 265         NULL,                           /* gss_inquire_cred_by_mech */
 266         glue_spnego_gss_inquire_names_for_mech,
 267         glue_spnego_gss_inquire_context,
 268         NULL,                           /* gss_internal_release_oid */
```



```
 269         glue_spnego_gss_wrap_size_limit,
 270         NULL, /* pname */
 271         NULL, /* userok */
 272         NULL, /* gss_export_name */
 277 /* EXPORT DELETE START */
 278 /* CRYPT DELETE START */
 279 #if 0
 280 /* CRYPT DELETE END */
 281         NULL, /* seal */
 282         NULL, /* unseal */
 283 /* CRYPT DELETE START */
 284 #endif
 285 /* CRYPT DELETE END */
 286 /* EXPORT DELETE END */
 273         NULL, /* sign */
 274         NULL, /* verify */
 275         NULL, /* gss_store_cred */
 276         spnego_gss_inquire_sec_context_by_oid, /* gss_inquire_sec_context_by_oid
 277 };
______unchanged_portion_omitted_
```

```
**********************************************************
    1329 Thu Jul 11 01:29:20 2013
new/usr/src/lib/libcrypt/Makefile
first pass
**********************************************************
```

```
#
# 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 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# ident "%Z%%M% %I%     %E% SMI"
#

include ../Makefile.lib

SUBDIRS=        $(MACH)
$(BUILD64)SUBDIRS += $(MACH64)

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

.KEEP_STATE:

all clean clobber install lint: $(SUBDIRS)

check: $(CHECKHDRS)

$(SUBDIRS):      FRC
        @cd $@; pwd; $(MAKE) $(TARGET)

FRC:

include ../Makefile.targ

# EXPORT DELETE START
EXPORT_SRC:
        $(RM) common/des.c+ common/des_crypt.c+ common/des_soft.c+ \
                common/des_decrypt.c+ common/des_encrypt.c+  \
                Makefile+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" < \
                common/des.c > common/des.c+
        $(MV) common/des.c+ common/des.c
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" < \
                common/des_crypt.c > common/des_crypt.c+
```



```
        $(MV) common/des_crypt.c+ common/des_crypt.c
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" < \
                common/des_soft.c > common/des_soft.c+
        $(MV) common/des_soft.c+ common/des_soft.c
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" < \
                common/des_encrypt.c > common/des_encrypt.c+
        $(MV) common/des_encrypt.c+ common/des_encrypt.c
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" < \
                common/des_decrypt.c > common/des_decrypt.c+
        $(MV) common/des_decrypt.c+ common/des_decrypt.c
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile common/des.c common/des_crypt.c \
                common/des_soft.c common/des_encrypt.c common/des_decrypt.c
# EXPORT DELETE END
```

```
**********************************************************
    4042 Thu Jul 11 01:29:21 2013
new/usr/src/lib/libcrypt/common/des.c
first pass
**********************************************************
______unchanged_portion_omitted_
  93 static int common_crypt(char *, char *, unsigned, unsigned, struct desparams *);

  95 /*
  96  * CBC mode encryption
  97  */
  98 int
  99 cbc_crypt(char *key, char *buf, size_t len, unsigned int mode, char *ivec)
 100 {
 101         int err = 0;

 103 /* EXPORT DELETE START */
 102         struct desparams dp;

 104         dp.des_mode = CBC;
 105         COPY8(ivec, dp.des_ivec);
 106         err = common_crypt(key, buf, len, mode, &dp);
 107         COPY8(dp.des_ivec, ivec);
 110 /* EXPORT DELETE END */
 108         return (err);
 109 }


 112 /*
 113  * ECB mode encryption
 114  */
 115 int
 116 ecb_crypt(char *key, char *buf, size_t len, unsigned int mode)
 117 {
 118         int ret = 0;

 123 /* EXPORT DELETE START */
 119         struct desparams dp;

 121         dp.des_mode = ECB;
 122         ret = common_crypt(key, buf, len, mode, &dp);
 128 /* EXPORT DELETE END */
 123         return (ret);
 124 }


 133 /* EXPORT DELETE START */
 127 /*
 128  * Common code to cbc_crypt() & ecb_crypt()
 129  */
 130 static int
 131 common_crypt(char *key, char *buf, unsigned len,
 132     unsigned mode, struct desparams *desp)
 133 {
 134         int desdev;
 135         int res;
 136         int g_desfd = UNOPENED;

 138         if ((len % 8) != 0 || len > DES_MAXDATA) {
 139                 return (DESERR_BADPARAM);
 140         }
 141         desp->des_dir =
 142             ((mode & DES_DIRMASK) == DES_ENCRYPT) ? ENCRYPT : DECRYPT;

 144         desdev = mode & DES_DEVMASK;
 145         COPY8(key, desp->des_key);
```



```
 146 #ifdef sun
 147         if (desdev == DES_HW) {
 148                 if (g_desfd < 0) {
 149                         if (g_desfd == -1 || (g_desfd = getdesfd()) < 0) {
 150                                 goto software;  /* no hardware device */
 151                         }
 152                 }

 154                 /*
 155                  * hardware
 156                  */
 157                 desp->des_len = len;
 158                 if (len <= DES_QUICKLEN) {
 159                         DESCOPY(buf, desp->des_data, len);
 160                         res = ioctl(g_desfd, (int)DESIOCQUICK, (char *)desp);
 161                         DESCOPY(desp->des_data, buf, len);
 162                 } else {
 163                         desp->des_buf = (uchar_t *)buf;
 164                         res = ioctl(g_desfd, (int)DESIOCBLOCK, (char *)desp);
 165                 }
 166                 return (res == 0 ? DESERR_NONE : DESERR_HWERROR);
 167         }
 168 software:
 169 #endif
 170         /*
 171          * software
 172          */
 173         if (!__des_crypt(buf, len, desp)) {
 174                 return (DESERR_HWERROR);
 175         }
 176         return (desdev == DES_SW ? DESERR_NONE : DESERR_NOHWDEVICE);
 177 }
 185 /* EXPORT DELETE END */
```

```
**********************************************************
    8758 Thu Jul 11 01:29:21 2013
new/usr/src/lib/libcrypt/common/des_crypt.c
first pass
**********************************************************
```

```
/*
 * 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 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*      Copyright (c) 1988 AT&T */
/*        All Rights Reserved   */

#pragma ident   "%Z%%M% %I%     %E% SMI"

#pragma weak _des_crypt = des_crypt
#pragma weak _des_encrypt = des_encrypt
#pragma weak _des_setkey = des_setkey

#include <sys/types.h>
#include <crypt.h>
#include "des_soft.h"

#include <stdlib.h>
#include <thread.h>
#include <pthread.h>
#include <sys/types.h>

/* EXPORT DELETE START */
/*
 * This program implements the
 * Proposed Federal Information Processing
 *  Data Encryption Standard.
 * See Federal Register, March 17, 1975 (40FR12134)
 */

/*
 * Initial permutation,
 */
static char IP[] = {
        58, 50, 42, 34, 26, 18, 10, 2,
        60, 52, 44, 36, 28, 20, 12, 4,
        62, 54, 46, 38, 30, 22, 14, 6,
        64, 56, 48, 40, 32, 24, 16, 8,
        57, 49, 41, 33, 25, 17, 9, 1,
```



```
        59, 51, 43, 35, 27, 19, 11, 3,
        61, 53, 45, 37, 29, 21, 13, 5,
        63, 55, 47, 39, 31, 23, 15, 7,
};
_____unchanged_portion_omitted_

/*
 * Set up the key schedule from the key.
 */

static mutex_t lock = DEFAULTMUTEX;

/* EXPORT DELETE END */


static void
des_setkey_nolock(const char *key)
{
/* EXPORT DELETE START */
        int i, j, k;
        char t;

        /*
         * First, generate C and D by permuting
         * the key.  The low order bit of each
         * 8-bit char is not used, so C and D are only 28
         * bits apiece.
         */
        for (i = 0; i < 28; i++) {
                C[i] = key[PC1_C[i]-1];
                D[i] = key[PC1_D[i]-1];
        }
        /*
         * To generate Ki, rotate C and D according
         * to schedule and pick up a permutation
         * using PC2.
         */
        for (i = 0; i < 16; i++) {
                /*
                 * rotate.
                 */
                for (k = 0; k < shifts[i]; k++) {
                        t = C[0];
                        for (j = 0; j < 28-1; j++)
                                C[j] = C[j+1];
                        C[27] = (char)t;
                        t = D[0];
                        for (j = 0; j < 28-1; j++)
                                D[j] = D[j+1];
                        D[27] = (char)t;
                }
                /*
                 * get Ki. Note C and D are concatenated.
                 */
                for (j = 0; j < 24; j++) {
                        KS[i][j] = C[PC2_C[j]-1];
                        KS[i][j+24] = D[PC2_D[j]-28-1];
                }
        }

        for (i = 0; i < 48; i++)
                E[i] = e2[i];
/* EXPORT DELETE END */
}

void
```

```
des_setkey(const char *key)
{
/* EXPORT DELETE START */
        (void) mutex_lock(&lock);
        des_setkey_nolock(key);
        (void) mutex_unlock(&lock);
/* EXPORT DELETE END */
}

/* EXPORT DELETE START */
/*
 * The 8 selection functions.
 * For some reason, they give a 0-origin
 * index, unlike everything else.
 */
static char S[8][64] = {
        14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7,
        0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8,
        4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0,
        15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13,

        15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10,
        3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5,
        0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15,
        13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9,

        10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8,
        13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1,
        13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7,
        1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12,

        7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15,
        13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9,
        10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4,
        3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14,

        2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9,
        14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6,
        4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14,
        11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3,

        12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11,
        10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8,
        9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6,
        4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13,

        4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1,
        13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6,
        1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2,
        6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12,

        13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7,
        1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2,
        7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8,
        2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11,
};
______unchanged_portion_omitted_

/*
 * The current block, divided into 2 halves.
 */
static char L[64];
static char tempL[32];
static char f[32];

/*
```



```
 * The combination of the key and the input, before selection.
 */
static char preS[48];

/*
 * The payoff: encrypt a block.
 */
/* EXPORT DELETE END */

static void
des_encrypt_nolock(char *block, int edflag)
{
/* EXPORT DELETE START */

        if (edflag)
                (void) _des_decrypt1(block, L, IP, &L[32],
                    preS, E, KS, S, f, tempL, P, FP);
        else
                (void) des_encrypt1(block, L, IP, &L[32],
                    preS, E, KS, S, f, tempL, P, FP);

/* EXPORT DELETE END */
}

void
des_encrypt(char *block, int edflag)
{
/* EXPORT DELETE START */
        (void) mutex_lock(&lock);
        des_encrypt_nolock(block, edflag);
        (void) mutex_unlock(&lock);
/* EXPORT DELETE END */
}
______unchanged_portion_omitted_

char *
des_crypt(const char *pw, const char *salt)
{
/* EXPORT DELETE START */
        int     i, j;
        char    c, temp;
        char block[66];
        static thread_key_t key = THR_ONCE_KEY;
        char *iobuf = _get_iobuf(&key, IOBUF_SIZE);

        (void) mutex_lock(&lock);
        for (i = 0; i < 66; i++)
                block[i] = 0;
        for (i = 0; (c = *pw) && (i < 64); pw++) {
                for (j = 0; j < 7; j++, i++)
                        block[i] = (c>>(6-j)) & 01;
                i++;
        }

        des_setkey_nolock(block);

        for (i = 0; i < 66; i++)
                block[i] = 0;

        for (i = 0; i < 2; i++) {
                c = *salt++;
                iobuf[i] = (char)c;
                if (c > 'Z')
                        c -= 6;
                if (c > '9')
                        c -= 7;
```

```
361                 c -= '.';
362                 for (j = 0; j < 6; j++) {
363                         if ((c>>j) & 01) {
364                                 temp = E[6*i+j];
365                                 E[6*i+j] = E[6*i+j+24];
366                                 E[6*i+j+24] = (char)temp;
367                         }
368                 }
369         }

371         for (i = 0; i < 25; i++)
372                 (void) des_encrypt_nolock(block, 0);

374         for (i = 0; i < 11; i++) {
375                 c = 0;
376                 for (j = 0; j < 6; j++) {
377                         c <<= 1;
378                         c |= block[6*i+j];
379                 }
380                 c += '.';
381                 if (c > '9')
382                         c += 7;
383                 if (c > 'Z')
384                         c += 6;
385                 iobuf[i+2] = (char)c;
386         }
387         iobuf[i+2] = 0;
388         if (iobuf[1] == 0)
389                 iobuf[1] = iobuf[0];
390         (void) mutex_unlock(&lock);
391         return (iobuf);
409 #if 0
410 /* EXPORT DELETE END */
411         return (0);
412 /* EXPORT DELETE START */
413 #endif
414 /* EXPORT DELETE END */
392 }
```

_____unchanged_portion_omitted_

```
**********************************************************
    2901 Thu Jul 11 01:29:22 2013
new/usr/src/lib/libcrypt/common/des_decrypt.c
first pass
**********************************************************
```

```
/*
 * 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 (c) 1988 AT&T */
/*        All Rights Reserved   */


/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma ident   "%Z%%M% %I%     %E% SMI"
/*LINTLIBRARY*/

#include <sys/types.h>

void
_des_decrypt1(char *block, char *L, char *IP, char *R, char *preS, char *E, char
{
/* EXPORT DELETE START */
        int     i, ii;
        int     t, j, k;
        char    t2;

        /*
         * First, permute the bits in the input
         */
        for (j = 0; j < 64; j++)
                L[j] = block[IP[j]-1];
        /*
         * Perform a decryption operation 16 times.
         */
        for (ii = 0; ii < 16; ii++) {
                i = 15-ii;
                /*
                 * Save the R array,
                 * which will be the new L.
                 */
                for (j = 0; j < 32; j++)
                        tempL[j] = R[j];
                /*
                 * Expand R to 48 bits using the E selector;
```



```
                 * exclusive-or with the current key bits.
                 */
                for (j = 0; j < 48; j++)
                        preS[j] = R[E[j]-1] ^ KS[i][j];
                /*
                 * The pre-select bits are now considered
                 * in 8 groups of 6 bits each.
                 * The 8 selection functions map these
                 * 6-bit quantities into 4-bit quantities
                 * and the results permuted
                 * to make an f(R, K).
                 * The indexing into the selection functions
                 * is peculiar; it could be simplified by
                 * rewriting the tables.
                 */
                for (j = 0; j < 8; j++) {
                        t = 6*j;
                        k = S[j][(preS[t+0]<<5)+
                                (preS[t+1]<<3)+
                                (preS[t+2]<<2)+
                                (preS[t+3]<<1)+
                                (preS[t+4]<<0)+
                                (preS[t+5]<<4)];
                        t = 4*j;
                        f[t+0] = (k>>3)&01;
                        f[t+1] = (k>>2)&01;
                        f[t+2] = (k>>1)&01;
                        f[t+3] = (k>>0)&01;
                }
                /*
                 * The new R is L ^ f(R, K).
                 * The f here has to be permuted first, though.
                 */
                for (j = 0; j < 32; j++)
                        R[j] = L[j] ^ f[P[j]-1];
                /*
                 * Finally, the new L (the original R)
                 * is copied back.
                 */
                for (j = 0; j < 32; j++)
                        L[j] = tempL[j];
        }
        /*
         * The output L and R are reversed.
         */
        for (j = 0; j < 32; j++) {
                t2 = L[j];
                L[j] = R[j];
                R[j] = t2;
        }
        /*
         * The final output
         * gets the inverse permutation of the very original.
         */
        for (j = 0; j < 64; j++)
                block[j] = L[FP[j]-1];
/* EXPORT DELETE END */
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    2902 Thu Jul 11 01:29:22 2013
new/usr/src/lib/libcrypt/common/des_encrypt.c
first pass
**********************************************************
/*
 * 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 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*      Copyright (c) 1988 AT&T */
/*        All Rights Reserved   */

#pragma ident   "%Z%%M% %I%     %E% SMI"

#pragma weak _des_encrypt1 = des_encrypt1

#include <sys/types.h>

void
des_encrypt1(char *block, char *L, char *IP, char *R, char *preS, char *E,
        char KS[][48], char S[][64], char *f, char *tempL, char *P, char *FP)
{
/* EXPORT DELETE START */
        int     i;
        int     t, j, k;
        char    t2;

        /*
         * First, permute the bits in the input
         */
        for (j = 0; j < 64; j++)
                L[j] = block[IP[j]-1];
        /*
         * Perform an encryption operation 16 times.
         */
        for (i = 0; i < 16; i++) {
                /*
                 * Save the R array,
                 * which will be the new L.
                 */
                for (j = 0; j < 32; j++)
                        tempL[j] = R[j];
                /*
                 * Expand R to 48 bits using the E selector;
```



```
                 * exclusive-or with the current key bits.
                 */
                for (j = 0; j < 48; j++)
                        preS[j] = R[E[j]-1] ^ KS[i][j];
                /*
                 * The pre-select bits are now considered
                 * in 8 groups of 6 bits each.
                 * The 8 selection functions map these
                 * 6-bit quantities into 4-bit quantities
                 * and the results permuted
                 * to make an f(R, K).
                 * The indexing into the selection functions
                 * is peculiar; it could be simplified by
                 * rewriting the tables.
                 */
                for (j = 0; j < 8; j++) {
                        t = 6*j;
                        k = S[j][(preS[t+0]<<5)+
                            (preS[t+1]<<3)+
                            (preS[t+2]<<2)+
                            (preS[t+3]<<1)+
                            (preS[t+4]<<0)+
                            (preS[t+5]<<4)];
                        t = 4*j;
                        f[t+0] = (k>>3)&01;
                        f[t+1] = (k>>2)&01;
                        f[t+2] = (k>>1)&01;
                        f[t+3] = (k>>0)&01;
                }
                /*
                 * The new R is L ^ f(R, K).
                 * The f here has to be permuted first, though.
                 */
                for (j = 0; j < 32; j++)
                        R[j] = L[j] ^ f[P[j]-1];
                /*
                 * Finally, the new L (the original R)
                 * is copied back.
                 */
                for (j = 0; j < 32; j++)
                        L[j] = tempL[j];
        }
        /*
         * The output L and R are reversed.
         */
        for (j = 0; j < 32; j++) {
                t2 = L[j];
                L[j] = R[j];
                R[j] = t2;
        }
        /*
         * The final output
         * gets the inverse permutation of the very original.
         */
        for (j = 0; j < 64; j++)
                block[j] = L[FP[j]-1];
/* EXPORT DELETE END */
}
______unchanged_portion_omitted_
```

```
**********************************************************
    9803 Thu Jul 11 01:29:23 2013
new/usr/src/lib/libcrypt/common/des_soft.c
first pass
**********************************************************
______unchanged_portion_omitted_
 119 #endif /* def _KERNEL */

 121 #ifdef CRYPT
 122 /*
 123  * Software encrypt or decrypt a block of data (multiple of 8 bytes)
 124  * Do the CBC ourselves if needed.
 125  */
 126 int
 127 __des_crypt(char *buf, unsigned int len, struct desparams *desp)
 128 {
 129 /* EXPORT DELETE START */
 129         short i;
 130         unsigned mode;
 131         unsigned dir;
 132         char nextiv[8];
 133         struct deskeydata softkey;

 135         mode = (unsigned)desp->des_mode;
 136         dir = (unsigned)desp->des_dir;
 137         des_setkey(desp->des_key, &softkey, dir);
 138         while (len != 0) {
 139                 switch (mode) {
 140                 case CBC:
 141                         switch (dir) {
 142                         case ENCRYPT:
 143                                 for (i = 0; i < 8; i++)
 144                                         buf[i] ^= desp->des_ivec[i];
 145                                 des_encrypt((uchar_t *)buf, &softkey);
 146                                 for (i = 0; i < 8; i++)
 147                                         desp->des_ivec[i] = buf[i];
 148                                 break;
 149                         case DECRYPT:
 150                                 for (i = 0; i < 8; i++)
 151                                         nextiv[i] = buf[i];
 152                                 des_encrypt((uchar_t *)buf, &softkey);
 153                                 for (i = 0; i < 8; i++) {
 154                                         buf[i] ^= desp->des_ivec[i];
 155                                         desp->des_ivec[i] = nextiv[i];
 156                                 }
 157                                 break;
 158                         }
 159                         break;
 160                 case ECB:
 161                         des_encrypt((uchar_t *)buf, &softkey);
 162                         break;
 163                 }
 164                 buf += 8;
 165                 len -= 8;
 166         }
 168 /* EXPORT DELETE END */
 167         return (1);
 168 }


 171 /*
 172  * Set the key and direction for an encryption operation
 173  * We build the 16 key entries here
 174  */
 175 static void
 176 des_setkey(uchar_t userkey[8], struct deskeydata *kd, unsigned int dir)
```



```
 177 {
 180 /* EXPORT DELETE START */
 178         long C, D;
 179         short i;

 181         /*
 182          * First, generate C and D by permuting
 183          * the key. The low order bit of each
 184          * 8-bit char is not used, so C and D are only 28
 185          * bits apiece.
 186          */
 187         {
 188                 short bit;
 189                 const short *pcc = PC1_C, *pcd = PC1_D;

 191                 C = D = 0;
 192                 for (i = 0; i < 28; i++) {
 193                         C <<= 1;
 194                         D <<= 1;
 195                         bit = *pcc++;
 196                         if (btst(userkey, bit))
 197                                 C |= 1;
 198                         bit = *pcd++;
 199                         if (btst(userkey, bit))
 200                                 D |= 1;
 201                 }
 202         }
 203         /*
 204          * To generate Ki, rotate C and D according
 205          * to schedule and pick up a permutation
 206          * using PC2.
 207          */
 208         for (i = 0; i < 16; i++) {
 209                 chunk_t *c;
 210                 short j, k, bit;
 211                 long bbit;

 213                 /*
 214                  * Do the "left shift" (rotate)
 215                  * We know we always rotate by either 1 or 2 bits
 216                  * the shifts table tells us if its 2
 217                  */
 218                 C <<= 1;
 219                 if (C & BIT28)
 220                         C |= 1;
 221                 D <<= 1;
 222                 if (D & BIT28)
 223                         D |= 1;
 224                 if (shifts[i]) {
 225                         C <<= 1;
 226                         if (C & BIT28)
 227                                 C |= 1;
 228                         D <<= 1;
 229                         if (D & BIT28)
 230                                 D |= 1;
 231                 }
 232                 /*
 233                  * get Ki. Note C and D are concatenated.
 234                  */
 235                 bit = 0;
 236                 switch (dir) {
 237                 case ENCRYPT:
 238                         c = &kd->keyval[i]; break;
 239                 case DECRYPT:
 240                         c = &kd->keyval[15 - i]; break;
 241                 }
```

```
242                 c->long0 = 0;
243                 c->long1 = 0;
244                 bbit = (1 << 5) << 24;
245                 for (j = 0; j < 4; j++) {
246                         for (k = 0; k < 6; k++) {
247                                 if (C & (BIT28 >> PC2_C[bit]))
248                                         c->long0 |= bbit >> k;
249                                 if (D & (BIT28 >> PC2_D[bit]))
250                                         c->long1 |= bbit >> k;
251                                 bit++;
252                         }
253                         bbit >>= 8;
254                 }

256         }
260 /* EXPORT DELETE END */
257 }



261 /*
262  * Do an encryption operation
263  * Much pain is taken (with preprocessor) to avoid loops so the compiler
264  * can do address arithmetic instead of doing it at runtime.
265  * Note that the byte-to-chunk conversion is necessary to guarantee
266  * processor byte-order independence.
267  */
268 static void
269 des_encrypt(uchar_t *data, struct deskeydata *kd)
270 {
275 /* EXPORT DELETE START */
271         chunk_t work1, work2;

273         /*
274          * Initial permutation
275          * and byte to chunk conversion
276          */
277         {
278                 const uint32_t *lp;
279                 uint32_t l0, l1, w;
280                 short i, pbit;

282                 work1.byte0 = data[0];
283                 work1.byte1 = data[1];
284                 work1.byte2 = data[2];
285                 work1.byte3 = data[3];
286                 work1.byte4 = data[4];
287                 work1.byte5 = data[5];
288                 work1.byte6 = data[6];
289                 work1.byte7 = data[7];
290                 l0 = l1 = 0;
291                 w = work1.long0;
292                 for (lp = &longtab[0], i = 0; i < 32; i++) {
293                         if (w & *lp++) {
294                                 pbit = IPtab[i];
295                                 if (pbit < 32)
296                                         l0 |= longtab[pbit];
297                                 else
298                                         l1 |= longtab[pbit-32];
299                         }
300                 }
301                 w = work1.long1;
302                 for (lp = &longtab[0], i = 32; i < 64; i++) {
303                         if (w & *lp++) {
304                                 pbit = IPtab[i];
305                                 if (pbit < 32)
```



```
306                                         l0 |= longtab[pbit];
307                                 else
308                                         l1 |= longtab[pbit-32];
309                         }
310                 }
311                 work2.long0 = l0;
312                 work2.long1 = l1;
313         }

315 /*
316  * Expand 8 bits of 32 bit R to 48 bit R
317  */
318 #define do_R_to_ER(op, b)       {                       \
319         const struct R_to_ER *p = &R_to_ER_tab[b][R.byte##b];   \
320         e0 op p->l0;                                    \
321         e1 op p->l1;                                    \
322 }
______unchanged_portion_omitted_

369         /*
370          * Apply the 16 ciphering steps
371          */
372         {
373                 uint32_t r0, l0, r1, l1;

375                 l0 = work2.long0;
376                 r0 = work2.long1;
377                 cipher(0, r0, l0, r1, l1);
378                 cipher(1, r1, l1, r0, l0);
379                 cipher(2, r0, l0, r1, l1);
380                 cipher(3, r1, l1, r0, l0);
381                 cipher(4, r0, l0, r1, l1);
382                 cipher(5, r1, l1, r0, l0);
383                 cipher(6, r0, l0, r1, l1);
384                 cipher(7, r1, l1, r0, l0);
385                 cipher(8, r0, l0, r1, l1);
386                 cipher(9, r1, l1, r0, l0);
387                 cipher(10, r0, l0, r1, l1);
388                 cipher(11, r1, l1, r0, l0);
389                 cipher(12, r0, l0, r1, l1);
390                 cipher(13, r1, l1, r0, l0);
391                 cipher(14, r0, l0, r1, l1);
392                 cipher(15, r1, l1, r0, l0);
393                 work1.long0 = r0;
394                 work1.long1 = l0;
395         }

397         /*
398          * Final permutation
399          * and chunk to byte conversion
400          */
401         {
402                 const uint32_t *lp;
403                 uint32_t l0, l1, w;
404                 short i, pbit;

406                 l0 = l1 = 0;
407                 w = work1.long0;
408                 for (lp = &longtab[0], i = 0; i < 32; i++) {
409                         if (w & *lp++) {
410                                 pbit = FPtab[i];
411                                 if (pbit < 32)
412                                         l0 |= longtab[pbit];
413                                 else
414                                         l1 |= longtab[pbit-32];
415                         }
```

```
416                 }
417                 w = work1.long1;
418                 for (lp = &longtab[0], i = 32; i < 64; i++) {
419                         if (w & *lp++) {
420                                 pbit = FPtab[i];
421                                 if (pbit < 32)
422                                         l0 |= longtab[pbit];
423                                 else
424                                         l1 |= longtab[pbit-32];
425                         }
426                 }
427                 work2.long0 = l0;
428                 work2.long1 = l1;
429         }
430         data[0] = work2.byte0;
431         data[1] = work2.byte1;
432         data[2] = work2.byte2;
433         data[3] = work2.byte3;
434         data[4] = work2.byte4;
435         data[5] = work2.byte5;
436         data[6] = work2.byte6;
437         data[7] = work2.byte7;

444 /* EXPORT DELETE END */
438 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1831 Thu Jul 11 01:29:24 2013
new/usr/src/lib/libgss/Makefile
first pass
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #
  25 # ident "%Z%%M% %I%     %E% SMI"
  26 #

  28 include ../Makefile.lib

  30 LIBRARY=        libgss.a

  32 # defines the duplicate sources we share with gsscred
  33 GSSCRED_DIR = $(SRC)/cmd/gss/gsscred
  34 DUPLICATE_SRC = gsscred_utils.c gsscred_file.c # gen_oids.c
  35 CLEAN_SRC = $(DUPLICATE_SRC) gen_oids.c

  37 SUBDIRS = $(MACH)
  38 $(BUILD64)SUBDIRS +=    $(MACH64)

  40 ROOTDIRS= $(ROOT)/usr/include
  41 GSSMECH_DIR=    $(ROOT)/usr/lib/gss

  43 all :=          TARGET= all
  44 clean :=        TARGET= clean
  45 clobber :=      TARGET= clobber
  46 install :=      TARGET= install
  47 lint :=         TARGET= lint

  49 POFILE =        $(LIBRARY:.a=.po)
  50 XGETFLAGS+=     -a
  51 MSGFILES =      `$(GREP) -l gettext *.[ch]`

  53 .KEEP_STATE:

  55 all clean clobber lint:  $(SUBDIRS)

  57 install: $(GSSMECH_DIR) all .WAIT $(SUBDIRS)

  59 check install_h:

  61 _msg:   $(MSGDOMAINPOFILE)
```



```
  63 $(POFILE):      pofile_MSGFILES

  65 $(GSSMECH_DIR):
  66         $(INS.dir)

  68 $(SUBDIRS):   FRC
  69         @cd $@; pwd; $(MAKE) $(TARGET)

  71 FRC:

  73 # include library targets
  74 include ../Makefile.targ

  76 # EXPORT DELETE START
  77 # Special target to clean up the source tree for export distribution
  78 # Warning: This target changes the source tree

  80 EXPORT_SRC:
  81         $(RM) Makefile+ g_seal.c+ g_unseal.c+
  82         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  83                 < g_seal.c > g_seal.c+
  84         $(MV) g_seal.c+ g_seal.c
  85         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  86                 < g_unseal.c > g_unseal.c+
  87         $(MV) g_unseal.c+ g_unseal.c
  88         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
  89                 < Makefile > Makefile+
  90         $(MV) Makefile+ Makefile
  91         $(CHMOD) 444 Makefile g_seal.c g_unseal.c

  93 # EXPORT DELETE END


  76 include $(SRC)/Makefile.msg.targ
```

```
**********************************************************
    4684 Thu Jul 11 01:29:24 2013
new/usr/src/lib/libgss/g_seal.c
first pass
**********************************************************
______unchanged_portion_omitted_

  68 /*ARGSUSED*/
  69 OM_uint32
  70 gss_seal(minor_status,
  71                 context_handle,
  72                 conf_req_flag,
  73                 qop_req,
  74                 input_message_buffer,
  75                 conf_state,
  76                 output_message_buffer)

  78 OM_uint32 *                     minor_status;
  79 gss_ctx_id_t                    context_handle;
  80 int                             conf_req_flag;
  81 int                             qop_req;
  82 gss_buffer_t                    input_message_buffer;
  83 int *                          conf_state;
  84 gss_buffer_t                    output_message_buffer;
  85 {
  86 /* EXPORT DELETE START */

  86         OM_uint32               status;
  87         gss_union_ctx_id_t      ctx;
  88         gss_mechanism           mech;

  90         status = val_seal_args(minor_status,
  91                         context_handle,
  92                         input_message_buffer,
  93                         output_message_buffer);
  94         if (status != GSS_S_COMPLETE)
  95                 return (status);

  97         /*
  98          * select the approprate underlying mechanism routine and
  99          * call it.
 100          */

 102         ctx = (gss_union_ctx_id_t) context_handle;
 103         mech = __gss_get_mechanism(ctx->mech_type);

 105         if (mech) {
 106                 if (mech->gss_seal) {
 107                         status = mech->gss_seal(
 108                                                 mech->context,
 109                                                 minor_status,
 110                                                 ctx->internal_ctx_id,
 111                                                 conf_req_flag,
 112                                                 qop_req,
 113                                                 input_message_buffer,
 114                                                 conf_state,
 115                                                 output_message_buffer);
 116                         if (status != GSS_S_COMPLETE)
 117                                 map_error(minor_status, mech);
 118                 } else
 119                         status = GSS_S_UNAVAILABLE;

 121                 return (status);
 122         }
 125 /* EXPORT DELETE END */
```



```
 124         return (GSS_S_BAD_MECH);
 125 }
______unchanged_portion_omitted_
```

```
**********************************************************
    2987 Thu Jul 11 01:29:25 2013
new/usr/src/lib/libgss/g_unseal.c
first pass
**********************************************************
```

```
/*
 * 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.
 */

/*
 *  glue routine gss_unseal
 */

#include <mechglueP.h>
#include "gssapiP_generic.h"

OM_uint32
gss_unseal(minor_status,
                context_handle,
                input_message_buffer,
                output_message_buffer,
                conf_state,
                qop_state)

OM_uint32 *             minor_status;
gss_ctx_id_t            context_handle;
gss_buffer_t            input_message_buffer;
gss_buffer_t            output_message_buffer;
int *                   conf_state;
int *                   qop_state;

{
/* EXPORT DELETE START */
        OM_uint32               status;
        gss_union_ctx_id_t      ctx;
        gss_mechanism           mech;

        if (minor_status != NULL)
                *minor_status = 0;

        if (output_message_buffer != GSS_C_NO_BUFFER) {
                output_message_buffer->length = 0;
                output_message_buffer->value = NULL;
        }

        if (minor_status == NULL)
```



```
                return (GSS_S_CALL_INACCESSIBLE_WRITE);

        if (context_handle == GSS_C_NO_CONTEXT)
                return (GSS_S_CALL_INACCESSIBLE_READ | GSS_S_NO_CONTEXT);

        if (input_message_buffer == GSS_C_NO_BUFFER ||
            GSS_EMPTY_BUFFER(input_message_buffer))
                return (GSS_S_CALL_INACCESSIBLE_READ);

        if (output_message_buffer == GSS_C_NO_BUFFER)
                return (GSS_S_CALL_INACCESSIBLE_WRITE);

        /*
         * select the approprate underlying mechanism routine and
         * call it.
         */

        ctx = (gss_union_ctx_id_t) context_handle;
        mech = __gss_get_mechanism(ctx->mech_type);

        if (mech) {
                if (mech->gss_unseal) {
                        status = mech->gss_unseal(
                                                mech->context,
                                                minor_status,
                                                ctx->internal_ctx_id,
                                                input_message_buffer,
                                                output_message_buffer,
                                                conf_state,
                                                qop_state);
                        if (status != GSS_S_COMPLETE)
                                map_error(minor_status, mech);
                } else
                        status = GSS_S_UNAVAILABLE;

                return (status);
        }

/* EXPORT DELETE END */

        return (GSS_S_BAD_MECH);
}
```

_____unchanged_portion_omitted_

****
19823 Thu Jul 11 01:29:25 2013
new/usr/src/lib/libldap5/sources/ldap/common/open.c
pass 2
****
_____***unchanged_portion_omitted_***

```
371 /*
372  * ldap_version - report version levels for important properties
373  * This function is deprecated.  Use ldap_get_option( ..., LDAP_OPT_API_INFO,
374  *      ... ) instead.
375  *
376  * Example:
377  *      LDAPVersion ver;
378  *      ldap_version( &ver );
379  *  if ( (ver.sdk_version < 100) || (ver.SSL_version < 300) )
380  *      fprintf( stderr, "LDAP SDK level insufficient\n" );
381  *
382  * or:
383  *  if ( ldap_version(NULL) < 100 )
384  *      fprintf( stderr, "LDAP SDK level insufficient\n" );
385  *
386  */

388 int
389 LDAP_CALL
390 ldap_version( LDAPVersion *ver )
391 {
392         if ( NULL != ver )
393         {
394                 memset( ver, 0, sizeof(*ver) );
395                 ver->sdk_version = (int)(VI_PRODUCTVERSION * 100);
396                 ver->protocol_version = LDAP_VERSION_MAX * 100;
397                 ver->SSL_version = SSL_VERSION * 100;
398                 /*
399                  * set security to none by default
400                  */

402                 ver->security_level = LDAP_SECURITY_NONE;
403 #if defined(LINK_SSL)
404 #if defined(NS_DOMESTIC)
404                 ver->security_level = 128;
406 #elif defined(NSS_EXPORT)
407                 ver->security_level = 40;
405 #endif
409 #endif

407         }
408         return (int)(VI_PRODUCTVERSION * 100);
409 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   24512 Thu Jul 11 01:29:26 2013
new/usr/src/lib/libldap5/sources/ldap/ssldap/clientinit.c
pass 2
**********************************************************
______unchanged_portion_omitted_


 157 static PRStatus local_SSLPLCY_Install(void)
 158 {
 159         return NSS_SetDomesticPolicy() ? PR_FAILURE : PR_SUCCESS;
 159         SECStatus s;

 161 #ifdef NS_DOMESTIC
 162         s = NSS_SetDomesticPolicy();
 163 #elif NS_EXPORT
 164         s = NSS_SetExportPolicy();
 165 #else
 166         s = PR_FAILURE;
 167 #endif
 168         return s?PR_FAILURE:PR_SUCCESS;
 160 }
______unchanged_portion_omitted_

 373 /*
 374  * Initialize ns/security so it can be used for SSL client authentication.
 375  * It is safe to call this more than once.
 376  *
 377  * If needkeydb == 0, no key database is opened and SSL server authentication
 378  * is supported but not client authentication.
 379  *
 380  * If "certdbpath" is NULL or "", the default cert. db is used (typically
 381  * ~/.netscape/cert7.db).
 382  *
 383  * If "certdbpath" ends with ".db" (case-insensitive compare), then
 384  * it is assumed to be a full path to the cert. db file; otherwise,
 385  * it is assumed to be a directory that contains a file called
 386  * "cert7.db" or "cert.db".
 387  *
 388  * If certdbhandle is non-NULL, it is assumed to be a pointer to a
 389  * SECCertDBHandle structure.  It is fine to pass NULL since this
 390  * routine will allocate one for you (CERT_GetDefaultDB() can be
 391  * used to retrieve the cert db handle).
 392  *
 393  * If "keydbpath" is NULL or "", the default key db is used (typically
 394  * ~/.netscape/key3.db).
 395  *
 396  * If "keydbpath" ends with ".db" (case-insensitive compare), then
 397  * it is assumed to be a full path to the key db file; otherwise,
 398  * it is assumed to be a directory that contains a file called
 399  * "key3.db"
 400  *
 401  * If certdbhandle is non-NULL< it is assumed to be a pointed to a
 402  * SECKEYKeyDBHandle structure.  It is fine to pass NULL since this
 403  * routine will allocate one for you (SECKEY_GetDefaultDB() can be
 404  * used to retrieve the cert db handle).
 405  */
 406 int
 407 LDAP_CALL
 408 ldapssl_clientauth_init( const char *certdbpath, void *certdbhandle,
 409     const int needkeydb, const char *keydbpath, void *keydbhandle )

 411 {
 412     int rc;
 413 #ifdef _SOLARIS_SDK
 414     char *enval;
```



```
 415     int rcenv = 0;
 416 #endif
 417
 418     /*
 419      *     LDAPDebug(LDAP_DEBUG_TRACE, "ldapssl_clientauth_init\n",0 ,0 ,0);
 420      */

 422     mutex_lock(&inited_mutex);
 423     if ( inited ) {
 424         mutex_unlock(&inited_mutex);
 425         return( 0 );
 426     }

 428     ldapssl_basic_init();

 430 #ifdef _SOLARIS_SDK
 431     if ((rcenv = update_nss_strict_fork_env(&enval)) == -1) {
 432         mutex_unlock(&inited_mutex);
 433         return (-1);
 434     }
 435 #endif

 437     /* Open the certificate database */
 438     rc = NSS_Init(certdbpath);
 439 #ifdef _SOLARIS_SDK
 440     /* Error from NSS_Init() more important! */
 441     if ((rcenv != 1) && (reset_nss_strict_fork_env(enval) != 0) && (rc == 0)) {
 442         ldapssl_free(&enval);
 443         mutex_unlock(&inited_mutex);
 444         return (-1);
 445     }
 446     ldapssl_free(&enval);
 447 #endif
 448     if (rc != 0) {
 449         if ((rc = PR_GetError()) >= 0)
 450             rc = -1;
 451         mutex_unlock(&inited_mutex);
 452         return (rc);
 453     }

 455     if (SSL_OptionSetDefault(SSL_ENABLE_SSL2, PR_FALSE)
 456             || SSL_OptionSetDefault(SSL_ENABLE_SSL3, PR_TRUE)) {
 457         if (( rc = PR_GetError()) >= 0 ) {
 458             rc = -1;
 459         }
 460         mutex_unlock(&inited_mutex);
 461         return( rc );
 462     }



 475 #if defined(NS_DOMESTIC)
 466     if (local_SSLPLCY_Install() == PR_FAILURE) {
 467       mutex_unlock(&inited_mutex);
 468       return( -1 );
 469     }
 480 #elif(NS_EXPORT)
 481     if (local_SSLPLCY_Install() == PR_FAILURE) {
 482       mutex_unlock(&inited_mutex);
 483       return( -1 );
 484     }
 485 #else
 486     mutex_unlock(&inited_mutex);
 487     return( -1 );
 488 #endif
```

```
471     inited = 1;
472     mutex_unlock(&inited_mutex);

474     return( 0 );

476 }

478 /*
479  * Initialize ns/security so it can be used for SSL client authentication.
480  * It is safe to call this more than once.
481  *
482  * If needkeydb == 0, no key database is opened and SSL server authentication
483  * is supported but not client authentication.
484  *
485  * If "certdbpath" is NULL or "", the default cert. db is used (typically
486  * ~/.netscape/cert7.db).
487  *
488  * If "certdbpath" ends with ".db" (case-insensitive compare), then
489  * it is assumed to be a full path to the cert. db file; otherwise,
490  * it is assumed to be a directory that contains a file called
491  * "cert7.db" or "cert.db".
492  *
493  * If certdbhandle is non-NULL, it is assumed to be a pointer to a
494  * SECCertDBHandle structure.  It is fine to pass NULL since this
495  * routine will allocate one for you (CERT_GetDefaultDB() can be
496  * used to retrieve the cert db handle).
497  *
498  * If "keydbpath" is NULL or "", the default key db is used (typically
499  * ~/.netscape/key3.db).
500  *
501  * If "keydbpath" ends with ".db" (case-insensitive compare), then
502  * it is assumed to be a full path to the key db file; otherwise,
503  * it is assumed to be a directory that contains a file called
504  * "key3.db"
505  *
506  * If certdbhandle is non-NULL< it is assumed to be a pointed to a
507  * SECKEYKeyDBHandle structure.  It is fine to pass NULL since this
508  * routine will allocate one for you (SECKEY_GetDefaultDB() can be
509  * used to retrieve the cert db handle).  */
510 int
511 LDAP_CALL
512 ldapssl_advclientauth_init(
513     const char *certdbpath, void *certdbhandle,
514     const int needkeydb, const char *keydbpath, void *keydbhandle,
515     const int needsecmoddb, const char *secmoddbpath,
516     const int sslstrength )
517 {
518     int rc;
519 #ifdef _SOLARIS_SDK
520     char *enval;
521     int rcenv = 0;
522 #endif

524     mutex_lock(&inited_mutex);
525     if ( inited ) {
526         mutex_unlock(&inited_mutex);
527         return( 0 );
528     }

530     /*
531      *    LDAPDebug(LDAP_DEBUG_TRACE, "ldapssl_advclientauth_init\n",0 ,0 ,0);
532      */

534     ldapssl_basic_init();

536 #ifdef _SOLARIS_SDK
```



```
537     if ((rcenv = update_nss_strict_fork_env(&enval)) == -1) {
538         mutex_unlock(&inited_mutex);
539         return (-1);
540     }
541 #endif

543     rc = NSS_Init(certdbpath);
544 #ifdef _SOLARIS_SDK
545     /* Error from NSS_Init() more important! */
546     if ((rcenv != 1) && (reset_nss_strict_fork_env(enval) != 0) && (rc == 0)) {
547         ldapssl_free(&enval);
548         mutex_unlock(&inited_mutex);
549         return (-1);
550     }
551     ldapssl_free(&enval);
552 #endif
553     if (rc != 0) {
554         if ((rc = PR_GetError()) >= 0)
555             rc = -1;
556         mutex_unlock(&inited_mutex);
557         return (rc);
558     }

579 #if defined(NS_DOMESTIC)
560     if (local_SSLPLCY_Install() == PR_FAILURE) {
561       mutex_unlock(&inited_mutex);
562       return( -1 );
563     }
584 #elif(NS_EXPORT)
585     if (local_SSLPLCY_Install() == PR_FAILURE) {
586       mutex_unlock(&inited_mutex);
587       return( -1 );
588     }
589 #else
590     mutex_unlock(&inited_mutex);
591     return( -1 );
592 #endif

565     inited = 1;
566     mutex_unlock(&inited_mutex);

568     return( ldapssl_set_strength( NULL, sslstrength));

570 }


573 /*
574  * Initialize ns/security so it can be used for SSL client authentication.
575  * It is safe to call this more than once.
576   */

578 /*
579  * XXXceb  This is a hack until the new IO functions are done.
580  * this function lives in ldapsinit.c
581  */
582 void set_using_pkcs_functions( int val );

584 int
585 LDAP_CALL
586 ldapssl_pkcs_init( const struct ldapssl_pkcs_fns *pfns )
587 {

589     char                *certdbName, *s, *keydbpath;
590     char                *certdbPrefix, *keydbPrefix;
591     char                *confDir, *keydbName;
592     static char         *secmodname =  "secmod.db";
```

```
593     int                 rc;
594 #ifdef _SOLARIS_SDK
595     char *enval;
596     int rcenv = 0;
597 #endif
598
599     mutex_lock(&inited_mutex);
600     if ( inited ) {
601         mutex_unlock(&inited_mutex);
602         return( 0 );
603     }
604 /*
605  * XXXceb  This is a hack until the new IO functions are done.
606  * this function MUST be called before ldap_enable_clienauth.
607  *
608  */
609     set_using_pkcs_functions( 1 );
610
611     /*
612      *    LDAPDebug(LDAP_DEBUG_TRACE, "ldapssl_pkcs_init\n",0 ,0 ,0);
613      */


616     ldapssl_basic_init();

618     pfns->pkcs_getcertpath( NULL, &s);
619     confDir = ldapssl_strdup( s );
620     certdbPrefix = ldapssl_strdup( s );
621     certdbName = ldapssl_strdup( s );
622     *certdbPrefix = 0;
623     splitpath(s, confDir, certdbPrefix, certdbName);

625     pfns->pkcs_getkeypath( NULL, &s);
626     keydbpath = ldapssl_strdup( s );
627     keydbPrefix = ldapssl_strdup( s );
628     keydbName = ldapssl_strdup( s );
629     *keydbPrefix = 0;
630     splitpath(s, keydbpath, keydbPrefix, keydbName);


633     /* verify confDir == keydbpath and adjust as necessary */
634     ldapssl_free((void **)&certdbName);
635     ldapssl_free((void **)&keydbName);
636     ldapssl_free((void **)&keydbpath);

638 #ifdef _SOLARIS_SDK
639     if ((rcenv = update_nss_strict_fork_env(&enval)) == -1) {
640         mutex_unlock(&inited_mutex);
641         return (-1);
642     }
643 #endif

645     rc = NSS_Initialize(confDir,certdbPrefix,keydbPrefix,secmodname,
646                 NSS_INIT_READONLY);

648     ldapssl_free((void **)&certdbPrefix);
649     ldapssl_free((void **)&keydbPrefix);
650     ldapssl_free((void **)&confDir);

652 #ifdef _SOLARIS_SDK
653     /* Error from NSS_Initialize() more important! */
654     if ((rcenv != 1) && (reset_nss_strict_fork_env(enval) != 0) && (rc == 0)) {
655         ldapssl_free(&enval);
656         mutex_unlock(&inited_mutex);
657         return (-1);
658     }
```



```
659     ldapssl_free(&enval);
660 #endif
661
662     if (rc != 0) {
663         if ((rc = PR_GetError()) >= 0)
664             rc = -1;
665         mutex_unlock(&inited_mutex);
666         return (rc);
667     }


670 #if 0   /* UNNEEDED BY LIBLDAP */
671     /* this is odd */
672     PK11_ConfigurePKCS11(NULL, NULL, tokDes, ptokDes, NULL, NULL, NULL, NULL, 0,
673 #endif  /* UNNEEDED BY LIBLDAP */

675     if (SSL_OptionSetDefault(SSL_ENABLE_SSL2, PR_FALSE)
676         || SSL_OptionSetDefault(SSL_ENABLE_SSL3, PR_TRUE)) {
677         if (( rc = PR_GetError()) >= 0 ) {
678             rc = -1;
679         }
680
681         mutex_unlock(&inited_mutex);
682         return( rc );
683     }
684
714 #if defined(NS_DOMESTIC)
685     if (local_SSLPLCY_Install() == PR_FAILURE) {
686       mutex_unlock(&inited_mutex);
687       return( -1 );
688     }
719 #elif(NS_EXPORT)
720     if (local_SSLPLCY_Install() == PR_FAILURE) {
721       mutex_unlock(&inited_mutex);
722       return( -1 );
723     }
724 #else
725     mutex_unlock(&inited_mutex);
726     return( -1 );
727 #endif

690     inited = 1;

692     if ( certdbName != NULL ) {
693         ldapssl_free((void **) &certdbName );
694     }
695
696     return( ldapssl_set_strength( NULL, LDAPSSL_AUTH_CNCHECK));
697 }
```

_____unchanged_portion_omitted_

```
**********************************************************
    3769 Thu Jul 11 01:29:27 2013
new/usr/src/lib/libnsl/Makefile
first pass
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#

PROTOCOL_DIR= $(ROOTHDRDIR)/rpcsvc
PROTOCOL_SRCDIR= $(SRC)/head/rpcsvc
PROTOCOL_UTS_SRCDIR= $(SRC)/uts/common/rpc

SUBDIRS =       $(MACH)
$(BUILD64)SUBDIRS += $(MACH64)

# objects are listed by source directory

# common utility code used in more than one directory
RPC_DERIVED_FILES=

GEN_DERIVED_FILES= \
        nis/gen/nis_clnt.h


PROTOCOL_FILES= \
        $(PROTOCOL_DIR)/daemon_utils.h  \
        $(PROTOCOL_DIR)/nis.x           \
        $(PROTOCOL_DIR)/nis.h           \
        $(PROTOCOL_DIR)/nis_object.x

PROTOCOL_FILES_UTS= \
        $(PROTOCOL_DIR)/key_prot.x

DERIVED_FILES= $(GEN_DERIVED_FILES) $(RPC_DERIVED_FILES)

#
# Make sure they get cleaned when necessary
#
CLEANFILES += $(DERIVED_FILES)

# include library definitions
include ../Makefile.lib
```



```
# header file delivered to /usr/include; internal to ON build process
HDRS =          nss.h
HDRDIR =        nss

LIBRARY= libnsl.a
TEXT_DOMAIN= SUNW_OST_NETRPC
POFILE= $(LIBRARY:.a=.po)
POFILES= generic.po _errlst.po

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


.KEEP_STATE:

all:            $(PROTOCOL_DIR) $(DERIVED_FILES) .WAIT $(SUBDIRS)

headers:        $(PROTOCOL_DIR) .WAIT $(PROTOCOL_FILES) $(PROTOCOL_FILES_UTS) \
                $(DERIVED_FILES)

install:        all .WAIT $(SUBDIRS)

install_h:      $(ROOTHDRS)

# nss.h isn't delivered; no reason to check
check:

clean clobber delete lint package:      $(SUBDIRS)

$(PROTOCOL_DIR):
        $(INS.dir)

$(PROTOCOL_DIR)/%.h:    $(PROTOCOL_SRCDIR)/%.h
        $(INS.file)

$(PROTOCOL_DIR)/nis.h:  $(PROTOCOL_SRCDIR)/nis.x $(PROTOCOL_SRCDIR)/nis_object.x
        $(RPCGEN) -C -h $(PROTOCOL_SRCDIR)/nis.x > nis-tmp.h
        $(SED) -e '/EDIT_START/,$$ d' < nis-tmp.h > nis.h
        $(RM) $@
        $(INS) -s -m $(FILEMODE) -f $(@D) nis.h
        $(RM) nis.h nis-tmp.h

$(PROTOCOL_DIR)/%.x:    $(PROTOCOL_SRCDIR)/%.x
        $(INS.file)

$(PROTOCOL_DIR)/%.x:    $(PROTOCOL_UTS_SRCDIR)/%.x
        $(INS.file)

#
# Rules for building the derived files
#
# Derived header files
#
nis/gen/nis_clnt.h: $(PROTOCOL_DIR)/nis.x $(PROTOCOL_DIR)/nis_object.x
        $(RPCGEN) -C -h $(PROTOCOL_DIR)/nis.x > nis_clnt-gen.h
        $(SED) -n -e '/EDIT_START/,$$ p' < nis_clnt-gen.h |\
        $(SED) -e 's/_3_svc/_svc/' |\
        $(SED) -e 's/_3/_clnt/' > $@
        $(RM) nis_clnt-gen.h
```

```
128 #
129 # Derived source files
130 #

132 # include library targets
133 include ../Makefile.targ

135 # EXPORT DELETE START
136 # CRYPT DELETE START
137 # Special target to clean up the source tree for export distribution
138 # Warning: This target changes the source tree
139 EXPORT_SRC:
140         $(RM) Makefile+ des/des_crypt.c+ des/des_soft.c+ key/xcrypt.c+
141         $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
142                 < des/des_crypt.c > des/des_crypt.c+
143         $(MV) des/des_crypt.c+ des/des_crypt.c
144         $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
145                 < des/des_soft.c > des/des_soft.c+
146         $(MV) des/des_soft.c+ des/des_soft.c
147         $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
148                 < key/xcrypt.c > key/xcrypt.c+
149         $(MV) key/xcrypt.c+ key/xcrypt.c
150         $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
151                 < Makefile > Makefile+
152         $(MV) Makefile+ Makefile
153         $(CHMOD) 444 Makefile des/des_crypt.c des/des_soft.c key/xcrypt.c

155 CRYPT_SRC:
156         $(RM) Makefile+
157         $(SED) -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d"      \
158                         < Makefile                                      \
159             | $(SED) -e "/EXPORT DELETE/d"                              \
160                         > Makefile+
161         $(MV) Makefile+ Makefile
162         $(CHMOD) 444 Makefile

164 # CRYPT DELETE END
165 # EXPORT DELETE END

135 _msg: $(MSGDOMAIN) $(POFILE)
136         $(RM) $(MSGDOMAIN)/$(POFILE)
137         $(CP) $(POFILE) $(MSGDOMAIN)

139 $(POFILE): $(DERIVED_FILES) .WAIT $(POFILES)
140         $(RM) $@
141         $(CAT) $(POFILES) > $@

143 _errlst.po:
144         $(RM) messages.po
145         $(XGETTEXT) -a nsl/_errlst.c
146         $(SED) -e '/^# msg/d' -e '/^domain/d' messages.po > $@
147         $(RM) messages.po

149 generic.po:
150         $(RM) messages.po
151         $(XGETTEXT) $(XGETFLAGS) `$(GREP) -l gettext */*.[ch] nis/*/*.[ch]*`
152         $(SED) -e '/^# msg/d' -e '/^domain/d' messages.po > $@
153         $(RM) messages.po

155 $(MSGDOMAIN):
156         $(INS.dir)

158 $(SUBDIRS):     FRC
159         @cd $@; pwd; $(MAKE) $(TARGET)

161 FRC:
```

```
**********************************************************
    4946 Thu Jul 11 01:29:27 2013
new/usr/src/lib/libnsl/des/des_crypt.c
first pass
**********************************************************
```

```
   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  */

  23 /*
  24  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  25  * Use is subject to license terms.
  26  */

  28 /* Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
  29 /* All Rights Reserved */

  31 /*
  32  * Portions of this source code were derived from Berkeley 4.3 BSD
  33  * under license from the Regents of the University of California.
  34  */

  36 #pragma ident   "%Z%%M% %I%     %E% SMI"

  38 /*
  39  * DES encryption library routines
  40  */

  42 #include "mt.h"
  43 #include <unistd.h>
  44 #include <fcntl.h>
  45 #include <sys/types.h>
  46 #include <rpc/des_crypt.h>
  47 /* EXPORT DELETE START */
  47 #ifdef sun
  48 #include <sys/ioctl.h>
  49 #include <sys/des.h>
  50 #define getdesfd()      (open("/dev/des", 0, 0))
  51 #else
  52 #include <des/des.h>
  53 #endif
  55 /* EXPORT DELETE END */
  54 #include <rpc/rpc.h>
  57 /* EXPORT DELETE START */

  56 extern int __des_crypt(char *, unsigned, struct desparams *);

  58 static int common_crypt(char *, char *, unsigned, unsigned, struct desparams *);
```



```
  60 /*
  61  * To see if chip is installed
  62  */
  63 #define UNOPENED (-2)
  64 static int g_desfd = UNOPENED;


  67 /*
  68  * Copy 8 bytes
  69  */
  70 #define COPY8(src, dst) { \
  71         char *a = (char *)dst; \
  72         char *b = (char *)src; \
  73         *a++ = *b++; *a++ = *b++; *a++ = *b++; *a++ = *b++; \
  74         *a++ = *b++; *a++ = *b++; *a++ = *b++; *a++ = *b++; \
  75 }
_____unchanged_portion_omitted_
  92 /* EXPORT DELETE END */

  90 /*
  91  * CBC mode encryption
  92  */
  93 int
  94 cbc_crypt(char *key, char *buf, size_t len, unsigned int mode, char *ivec)
  95 {
 100 /* EXPORT DELETE START */
  96         int err;
  97         struct desparams dp;

  99         dp.des_mode = CBC;
 100         COPY8(ivec, dp.des_ivec);
 101         err = common_crypt(key, buf, len, mode, &dp);
 102         COPY8(dp.des_ivec, ivec);
 103         return (err);
 109 #if 0
 110 /* EXPORT DELETE END */
 111         return (DESERR_HWERROR);
 112 /* EXPORT DELETE START */
 113 #endif
 114 /* EXPORT DELETE END */
 104 }


 107 /*
 108  * ECB mode encryption
 109  */
 110 int
 111 ecb_crypt(char *key, char *buf, size_t len, unsigned int mode)
 112 {
 124 /* EXPORT DELETE START */
 113         struct desparams dp;

 115         dp.des_mode = ECB;
 116         return (common_crypt(key, buf, len, mode, &dp));
 129 #if 0
 130 /* EXPORT DELETE END */
 131         return (DESERR_HWERROR);
 132 /* EXPORT DELETE START */
 133 #endif
 134 /* EXPORT DELETE END */
 117 }


 138 /* EXPORT DELETE START */
```

```
121 /*
122  * Common code to cbc_crypt() & ecb_crypt()
123  */
124 static int
125 common_crypt(char *key, char *buf, unsigned len, unsigned mode,
126                                                 struct desparams *desp)
127 {
128         int desdev;
129         int res;

131         if ((len % 8) != 0 || len > DES_MAXDATA)
132                 return (DESERR_BADPARAM);
133         desp->des_dir =
134                 ((mode & DES_DIRMASK) == DES_ENCRYPT) ? ENCRYPT : DECRYPT;

136         desdev = mode & DES_DEVMASK;
137         COPY8(key, desp->des_key);
138 #ifdef sun
139         if (desdev == DES_HW) {
140                 if (g_desfd < 0) {
141                         if (g_desfd == -1 || (g_desfd = getdesfd()) < 0) {
142                                 goto software;  /* no hardware device */
143                         }
144                 }

146                 /*
147                  * hardware
148                  */
149                 desp->des_len = len;
150                 if (len <= DES_QUICKLEN) {
151                         DESCOPY(buf, desp->des_data, len);
152                         res = ioctl(g_desfd, DESIOCQUICK, (char *)desp);
153                         DESCOPY(desp->des_data, buf, len);
154                 } else {
155                         desp->des_buf = (uchar_t *)buf;
156                         res = ioctl(g_desfd, DESIOCBLOCK, (char *)desp);
157                 }
158                 return (res == 0 ? DESERR_NONE : DESERR_HWERROR);
159         }
160 software:
161 #endif
162         /*
163          * software
164          */
165         if (!__des_crypt(buf, len, desp))
166                 return (DESERR_HWERROR);
167         return (desdev == DES_SW ? DESERR_NONE : DESERR_NOHWDEVICE);
168 }
188 /* EXPORT DELETE END */

190 /* EXPORT DELETE START */
170 static int
171 desN_crypt(des_block keys[], int keynum, char *buf, unsigned int len,
172                 unsigned int mode, char *ivec)
173 {
174         unsigned int m = mode & (DES_ENCRYPT | DES_DECRYPT);
175         unsigned int flags = mode & ~(DES_ENCRYPT | DES_DECRYPT);
176         des_block svec, dvec;
177         int i, j, stat;

179         if (keynum < 1)
180                 return (DESERR_BADPARAM);

182         (void) memcpy(svec.c, ivec, sizeof (des_block));
183         for (i = 0; i < keynum; i++) {
184                 j = (mode & DES_DECRYPT) ? keynum - 1 - i : i;
```



```
185                 stat = cbc_crypt(keys[j].c, buf, len, m | flags, ivec);
186                 if (mode & DES_DECRYPT && i == 0)
187                         (void) memcpy(dvec.c, ivec, sizeof (des_block));

189                 if (DES_FAILED(stat))
190                         return (stat);

192                 m = (m == DES_ENCRYPT ? DES_DECRYPT : DES_ENCRYPT);

194                 if ((mode & DES_DECRYPT) || i != keynum - 1 || i%2)
195                         (void) memcpy(ivec, svec.c, sizeof (des_block));
196         }
197         if (keynum % 2 == 0)
198                 stat = cbc_crypt(keys[0].c, buf, len, mode, ivec);

200         if (mode & DES_DECRYPT)
201                 (void) memcpy(ivec, dvec.c, sizeof (des_block));

203         return (stat);
204 }
226 /* EXPORT DELETE END */



208 int
209 __cbc_triple_crypt(des_block keys[], char *buf,  uint_t len,
210                         uint_t mode, char *ivec)
211 {
234 /* EXPORT DELETE START */
212         return (desN_crypt(keys, 3, buf, len, mode, ivec));
236 #if 0
237 /* EXPORT DELETE END */
238         return (DESERR_HWERROR);
239 /* EXPORT DELETE START */
240 #endif
241 /* EXPORT DELETE END */
213 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   11485 Thu Jul 11 01:29:28 2013
new/usr/src/lib/libnsl/des/des_soft.c
first pass
**********************************************************
______unchanged_portion_omitted_

 158 /*
 159  * Software encrypt or decrypt a block of data (multiple of 8 bytes)
 160  * Do the CBC ourselves if needed.
 161  */
 162 int
 163 __des_crypt(char *buf, unsigned len, struct desparams *desp)
 164 {
 165 /* EXPORT DELETE START */
 165         short i;
 166         unsigned mode;
 167         unsigned dir;
 168         char nextiv[8];
 169         struct deskeydata softkey;

 171         mode = (unsigned)desp->des_mode;
 172         dir = (unsigned)desp->des_dir;
 173         (void) __des_setkey(desp->des_key, &softkey, dir);
 174         while (len != 0) {
 175                 switch (mode) {
 176                 case CBC:
 177                         switch (dir) {
 178                         case ENCRYPT:
 179                                 for (i = 0; i < 8; i++)
 180                                         buf[i] ^= desp->des_ivec[i];
 181                                 (void) __des_encrypt((uchar_t *)buf, &softkey);
 182                                 for (i = 0; i < 8; i++)
 183                                         desp->des_ivec[i] = buf[i];
 184                                 break;
 185                         case DECRYPT:
 186                                 for (i = 0; i < 8; i++)
 187                                         nextiv[i] = buf[i];
 188                                 (void) __des_encrypt((uchar_t *)buf, &softkey);
 189                                 for (i = 0; i < 8; i++) {
 190                                         buf[i] ^= desp->des_ivec[i];
 191                                         desp->des_ivec[i] = nextiv[i];
 192                                 }
 193                                 break;
 194                         }
 195                         break;
 196                 case ECB:
 197                         (void) __des_encrypt((uchar_t *)buf, &softkey);
 198                         break;
 199                 }
 200                 buf += 8;
 201                 len -= 8;
 202         }
 204 /* EXPORT DELETE END */
 203         return (1);
 204 }


 207 /*
 208  * Set the key and direction for an encryption operation
 209  * We build the 16 key entries here
 210  */
 211 static int
 212 __des_setkey(uchar_t userkey[8], struct deskeydata *kd, unsigned dir)
 213 {
 216 /* EXPORT DELETE START */
```



```
 214         int32_t C, D;
 215         short i;

 217         /*
 218          * First, generate C and D by permuting
 219          * the key. The low order bit of each
 220          * 8-bit char is not used, so C and D are only 28
 221          * bits apiece.
 222          */
 223         {
 224                 short bit;
 225                 const short *pcc = PC1_C, *pcd = PC1_D;

 227                 C = D = 0;
 228                 for (i = 0; i < 28; i++) {
 229                         C <<= 1;
 230                         D <<= 1;
 231                         bit = *pcc++;
 232                         if (btst(userkey, bit))
 233                                 C |= 1;
 234                         bit = *pcd++;
 235                         if (btst(userkey, bit))
 236                                 D |= 1;
 237                 }
 238         }
 239         /*
 240          * To generate Ki, rotate C and D according
 241          * to schedule and pick up a permutation
 242          * using PC2.
 243          */
 244         for (i = 0; i < 16; i++) {
 245                 chunk_t *c;
 246                 short j, k, bit;
 247                 uint32_t bbit;

 249                 /*
 250                  * Do the "left shift" (rotate)
 251                  * We know we always rotate by either 1 or 2 bits
 252                  * the shifts table tells us if its 2
 253                  */
 254                 C <<= 1;
 255                 if (C & BIT28)
 256                         C |= 1;
 257                 D <<= 1;
 258                 if (D & BIT28)
 259                         D |= 1;
 260                 if (shifts[i]) {
 261                         C <<= 1;
 262                         if (C & BIT28)
 263                                 C |= 1;
 264                         D <<= 1;
 265                         if (D & BIT28)
 266                                 D |= 1;
 267                 }
 268                 /*
 269                  * get Ki. Note C and D are concatenated.
 270                  */
 271                 bit = 0;
 272                 switch (dir) {
 273                 case ENCRYPT:
 274                         c = &kd->keyval[i]; break;
 275                 case DECRYPT:
 276                         c = &kd->keyval[15 - i]; break;
 277                 }
 278                 c->long0 = 0;
 279                 c->long1 = 0;
```

```
280                 bbit = (1 << 5) << 24;
281                 for (j = 0; j < 4; j++) {
282                         for (k = 0; k < 6; k++) {
283                                 if (C & (BIT28 >> PC2_C[bit]))
284                                         c->long0 |= bbit >> k;
285                                 if (D & (BIT28 >> PC2_D[bit]))
286                                         c->long1 |= bbit >> k;
287                                 bit++;
288                         }
289                         bbit >>= 8;
290                 }

292         }
296 /* EXPORT DELETE END */
293         return (1);
294 }



298 /*
299  * Do an encryption operation
300  * Much pain is taken (with preprocessor) to avoid loops so the compiler
301  * can do address arithmetic instead of doing it at runtime.
302  * Note that the byte-to-chunk conversion is necessary to guarantee
303  * processor byte-order independence.
304  */
305 static int
306 __des_encrypt(uchar_t *data, struct deskeydata *kd)
307 {
312 /* EXPORT DELETE START */
308         chunk_t work1, work2;

310         /*
311          * Initial permutation
312          * and byte to chunk conversion
313          */
314         {
315                 const uint32_t *lp;
316                 uint32_t l0, l1, w;
317                 short i, pbit;

319                 work1.byte0 = data[0];
320                 work1.byte1 = data[1];
321                 work1.byte2 = data[2];
322                 work1.byte3 = data[3];
323                 work1.byte4 = data[4];
324                 work1.byte5 = data[5];
325                 work1.byte6 = data[6];
326                 work1.byte7 = data[7];
327                 l0 = l1 = 0;
328                 w = work1.long0;
329                 for (lp = (uint32_t *)&longtab[0], i = 0; i < 32; i++) {
330                         if (w & *lp++) {
331                                 pbit = IPtab[i];
332                                 if (pbit < 32)
333                                         l0 |= longtab[pbit];
334                                 else
335                                         l1 |= longtab[pbit-32];
336                         }
337                 }
338                 w = work1.long1;
339                 for (lp = (uint32_t *)&longtab[0], i = 32; i < 64; i++) {
340                         if (w & *lp++) {
341                                 pbit = IPtab[i];
342                                 if (pbit < 32)
343                                         l0 |= longtab[pbit];
```



```
344                                 else
345                                         l1 |= longtab[pbit-32];
346                         }
347                 }
348                 work2.long0 = l0;
349                 work2.long1 = l1;
350         }

352 /*
353  * Expand 8 bits of 32 bit R to 48 bit R
354  */
355 #define do_R_to_ER(op, b)       {                               \
356         const struct R_to_ER *p = &R_to_ER_tab[b][R.byte##b];   \
357         e0 op p->l0;                                            \
358         e1 op p->l1;                                            \
359 }
_____unchanged_portion_omitted_

404         /*
405          * Apply the 16 ciphering steps
406          */
407         {
408                 uint32_t r0, l0, r1, l1;

410                 l0 = work2.long0;
411                 r0 = work2.long1;
412                 cipher(0, r0, l0, r1, l1);
413                 cipher(1, r1, l1, r0, l0);
414                 cipher(2, r0, l0, r1, l1);
415                 cipher(3, r1, l1, r0, l0);
416                 cipher(4, r0, l0, r1, l1);
417                 cipher(5, r1, l1, r0, l0);
418                 cipher(6, r0, l0, r1, l1);
419                 cipher(7, r1, l1, r0, l0);
420                 cipher(8, r0, l0, r1, l1);
421                 cipher(9, r1, l1, r0, l0);
422                 cipher(10, r0, l0, r1, l1);
423                 cipher(11, r1, l1, r0, l0);
424                 cipher(12, r0, l0, r1, l1);
425                 cipher(13, r1, l1, r0, l0);
426                 cipher(14, r0, l0, r1, l1);
427                 cipher(15, r1, l1, r0, l0);
428                 work1.long0 = r0;
429                 work1.long1 = l0;
430         }

432         /*
433          * Final permutation
434          * and chunk to byte conversion
435          */
436         {
437                 uint32_t *lp;
438                 uint32_t l0, l1, w;
439                 short i, pbit;

441                 l0 = l1 = 0;
442                 w = work1.long0;
443                 for (lp = (uint32_t *)&longtab[0], i = 0; i < 32; i++) {
444                         if (w & *lp++) {
445                                 pbit = FPtab[i];
446                                 if (pbit < 32)
447                                         l0 |= longtab[pbit];
448                                 else
449                                         l1 |= longtab[pbit-32];
450                         }
451                 }
```

```
452                 w = work1.long1;
453                 for (lp = (uint32_t *)&longtab[0], i = 32; i < 64; i++) {
454                         if (w & *lp++) {
455                                 pbit = FPtab[i];
456                                 if (pbit < 32)
457                                         l0 |= longtab[pbit];
458                                 else
459                                         l1 |= longtab[pbit-32];
460                         }
461                 }
462                 work2.long0 = l0;
463                 work2.long1 = l1;
464         }
465         data[0] = work2.byte0;
466         data[1] = work2.byte1;
467         data[2] = work2.byte2;
468         data[3] = work2.byte3;
469         data[4] = work2.byte4;
470         data[5] = work2.byte5;
471         data[6] = work2.byte6;
472         data[7] = work2.byte7;

479 /* EXPORT DELETE END */
474         return (1);
475 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   13507 Thu Jul 11 01:29:28 2013
new/usr/src/lib/libnsl/key/xcrypt.c
first pass
**********************************************************
```

```
   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  */

  23 /*
  24  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  25  * Use is subject to license terms.
  26  */

  28 /* Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
  29 /* All Rights Reserved */

  31 /*
  32  * Portions of this source code were derived from Berkeley 4.3 BSD
  33  * under license from the Regents of the University of California.
  34  */

  36 #pragma ident   "%Z%%M% %I%     %E% SMI"

  38 /*
  39  * Hex encryption/decryption and utility routines
  40  */

  42 #include "mt.h"
  43 #include <stdio.h>
  44 #include <stdlib.h>
  45 #include <sys/types.h>
  46 #include <rpc/rpc.h>
  47 #include <rpc/key_prot.h>   /* for KEYCHECKSUMSIZE */
  48 #include <rpc/des_crypt.h>
  49 #include <string.h>
  50 #include <rpcsvc/nis_dhext.h>
  51 #include <md5.h>

  53 #define MD5HEXSIZE      32

  55 extern int bin2hex(int len, unsigned char *binnum, char *hexnum);
  56 extern int hex2bin(int len, char *hexnum, char *binnum);
  57 static char hex[];      /* forward */
  58 static char hexval();

  60 int passwd2des(char *, char *);
  61 static int weak_DES_key(des_block);
```



```
  63 /* EXPORT DELETE START */
  63 /*
  64  * For export control reasons, we want to limit the maximum size of
  65  * data that can be encrypted or decrypted.  We limit this to 1024
  66  * bits of key data, which amounts to 128 bytes.
  67  *
  68  * For the extended DH project, we have increased it to
  69  * 144 bytes (128key + 16checksum) to accomadate all the 128 bytes
  70  * being used by the new 1024bit keys plus 16 bytes MD5 checksum.
  71  * We discussed this with Sun's export control office and lawyers
  72  * and we have reason to believe this is ok for export.
  73  */
  74 #define MAX_KEY_CRYPT_LEN       144
  76 /* EXPORT DELETE END */

  76 /*
  77  * Encrypt a secret key given passwd
  78  * The secret key is passed and returned in hex notation.
  79  * Its length must be a multiple of 16 hex digits (64 bits).
  80  */
  81 int
  82 xencrypt(secret, passwd)
  83         char *secret;
  84         char *passwd;
  85 {
  88 /* EXPORT DELETE START */
  86         char key[8];
  87         char ivec[8];
  88         char *buf;
  89         int err;
  90         int len;

  92         len = (int)strlen(secret) / 2;
  93         if (len > MAX_KEY_CRYPT_LEN)
  94                 return (0);
  95         buf = malloc((unsigned)len);
  96         (void) hex2bin(len, secret, buf);
  97         (void) passwd2des(passwd, key);
  98         (void) memset(ivec, 0, 8);

 100         err = cbc_crypt(key, buf, len, DES_ENCRYPT | DES_HW, ivec);
 101         if (DES_FAILED(err)) {
 102                 free(buf);
 103                 return (0);
 104         }
 105         (void) bin2hex(len, (unsigned char *) buf, secret);
 106         free(buf);
 107         return (1);
 111 #if 0
 112 /* EXPORT DELETE END */
 113         return (0);
 114 /* EXPORT DELETE START */
 115 #endif
 116 /* EXPORT DELETE END */
 108 }

 110 /*
 111  * Decrypt secret key using passwd
 112  * The secret key is passed and returned in hex notation.
 113  * Once again, the length is a multiple of 16 hex digits
 114  */
 115 int
 116 xdecrypt(secret, passwd)
 117         char *secret;
 118         char *passwd;
```

```
119 {
129 /* EXPORT DELETE START */
120         char key[8];
121         char ivec[8];
122         char *buf;
123         int err;
124         int len;

126         len = (int)strlen(secret) / 2;
127         if (len > MAX_KEY_CRYPT_LEN)
128                 return (0);
129         buf = malloc((unsigned)len);

131         (void) hex2bin(len, secret, buf);
132         (void) passwd2des(passwd, key);
133         (void) memset(ivec, 0, 8);

135         err = cbc_crypt(key, buf, len, DES_DECRYPT | DES_HW, ivec);
136         if (DES_FAILED(err)) {
137                 free(buf);
138                 return (0);
139         }
140         (void) bin2hex(len, (unsigned char *) buf, secret);
141         free(buf);
142         return (1);
153 #if 0
154 /* EXPORT DELETE END */
155         return (0);
156 /* EXPORT DELETE START */
157 #endif
158 /* EXPORT DELETE END */
143 }
```

unchanged_portion_omitted_

```
223 /*
224  * Generic key length/algorithm version of xencrypt().
225  *
226  * Encrypt a secret key given passwd.
227  * The secret key is passed in hex notation.
228  * Arg encrypted_secret will be set to point to the encrypted
229  * secret key (NUL term, hex notation).
230  *
231  * Its length must be a multiple of 16 hex digits (64 bits).
232  *
233  * For 192-0 (AUTH_DES), then encrypt using the same method as xencrypt().
234  *
235  * If arg do_chksum is TRUE, append the checksum before the encrypt.
236  * For 192-0, the checksum is done the same as in xencrypt().  For
237  * bigger keys, MD5 is used.
238  *
239  * Arg netname can be NULL for 192-0.
240  */
241 int
242 xencrypt_g(
243         char *secret,                   /* in  */
244         keylen_t keylen,                /* in  */
245         algtype_t algtype,              /* in  */
246         const char *passwd,             /* in  */
247         const char netname[],           /* in  */
248         char **encrypted_secret,        /* out */
249         bool_t do_chksum)               /* in  */
250 {
267 /* EXPORT DELETE START */
251         des_block key;
252         char ivec[8];
253         char *binkeybuf;
```



```
254         int err;
255         const int classic_des = keylen == 192 && algtype == 0;
256         const int hexkeybytes = BITS2NIBBLES(keylen);
257         const int keychecksumsize = classic_des ? KEYCHECKSUMSIZE : MD5HEXSIZE;
258         const int binkeybytes = do_chksum ? keylen/8 + keychecksumsize/2 :
259                 keylen/8;
260         const int bufsize = do_chksum ? hexkeybytes + keychecksumsize + 1 :
261                 hexkeybytes + 1;
262         char *hexkeybuf;

264         if (!secret || !keylen || !passwd || !encrypted_secret)
265                 return (0);

267         if ((hexkeybuf = malloc(bufsize)) == 0)
268                 return (0);

270         (void) memcpy(hexkeybuf, secret, hexkeybytes);
271         if (do_chksum)
272                 if (classic_des) {
273                         (void) memcpy(hexkeybuf + hexkeybytes, secret,
274                                 keychecksumsize);
275                 } else {
276                         MD5_CTX md5_ctx;
277                         char md5hexbuf[MD5HEXSIZE + 1] = {0};
278                         uint8_t digest[MD5HEXSIZE/2];

280                         MD5Init(&md5_ctx);
281                         MD5Update(&md5_ctx, (unsigned char *)hexkeybuf,
282                                 hexkeybytes);
283                         MD5Final(digest, &md5_ctx);

285                         /* convert md5 binary digest to hex */
286                         (void) bin2hex(MD5HEXSIZE/2, digest, md5hexbuf);

288                         /* append the hex md5 string to the end of the key */
289                         (void) memcpy(hexkeybuf + hexkeybytes,
290                                 (void *)md5hexbuf, MD5HEXSIZE);
291                 }
292         hexkeybuf[bufsize - 1] = 0;

294         if (binkeybytes > MAX_KEY_CRYPT_LEN) {
295                 free(hexkeybuf);
296                 return (0);
297         }
298         if ((binkeybuf = malloc((unsigned)binkeybytes)) == 0) {
299                 free(hexkeybuf);
300                 return (0);
301         }

303         (void) hex2bin(binkeybytes, hexkeybuf, binkeybuf);
304         if (classic_des)
305                 (void) passwd2des((char *)passwd, key.c);
306         else
307                 if (netname)
308                         (void) passwd2des_g(passwd, netname,
309                                 (int)strlen(netname), &key, FALSE);
310                 else {
311                         free(hexkeybuf);
312                         return (0);
313                 }

315         (void) memset(ivec, 0, 8);

317         err = cbc_crypt(key.c, binkeybuf, binkeybytes, DES_ENCRYPT | DES_HW,
318                         ivec);
319         if (DES_FAILED(err)) {
```

```
320                 free(hexkeybuf);
321                 free(binkeybuf);
322                 return (0);
323         }
324         (void) bin2hex(binkeybytes, (unsigned char *) binkeybuf, hexkeybuf);
325         free(binkeybuf);
326         *encrypted_secret = hexkeybuf;
327         return (1);
345 #if 0
346 /* EXPORT DELETE END */
347         return (0);
348 /* EXPORT DELETE START */
349 #endif
350 /* EXPORT DELETE END */
328 }

330 /*
331  * Generic key len and alg type for version of xdecrypt.
332  *
333  * Decrypt secret key using passwd.  The decrypted secret key
334  * *overwrites* the supplied encrypted secret key.
335  * The secret key is passed and returned in hex notation.
336  * Once again, the length is a multiple of 16 hex digits.
337  *
338  * If 'do_chksum' is TRUE, the 'secret' buffer is assumed to contain
339  * a checksum calculated by a call to xencrypt_g().
340  *
341  * If keylen is 192 and algtype is 0, then decrypt the same way
342  * as xdecrypt().
343  *
344  * Arg netname can be NULL for 192-0.
345  */
346 int
347 xdecrypt_g(
348         char *secret,           /* out  */
349         int keylen,             /* in  */
350         int algtype,            /* in  */
351         const char *passwd,     /* in  */
352         const char netname[],   /* in  */
353         bool_t do_chksum)       /* in  */
354 {
378 /* EXPORT DELETE START */
355         des_block key;
356         char ivec[8];
357         char *buf;
358         int err;
359         int len;
360         const int classic_des = keylen == 192 && algtype == 0;
361         const int hexkeybytes = BITS2NIBBLES(keylen);
362         const int keychecksumsize = classic_des ? KEYCHECKSUMSIZE : MD5HEXSIZE;

364         len = (int)strlen(secret) / 2;
365         if (len > MAX_KEY_CRYPT_LEN)
366                 return (0);
367         if ((buf = malloc((unsigned)len)) == 0)
368                 return (0);

370         (void) hex2bin(len, secret, buf);
371         if (classic_des)
372                 (void) passwd2des((char *)passwd, key.c);
373         else
374                 if (netname)
375                         (void) passwd2des_g(passwd, netname,
376                                         (int)strlen(netname), &key, FALSE);
377                 else {
378                         free(buf);
```



```
379                         return (0);
380                 }
381         (void) memset(ivec, 0, 8);

383         err = cbc_crypt(key.c, buf, len, DES_DECRYPT | DES_HW, ivec);
384         if (DES_FAILED(err)) {
385                 free(buf);
386                 return (0);
387         }
388         (void) bin2hex(len, (unsigned char *) buf, secret);
389         free(buf);

391         if (do_chksum)
392                 if (classic_des) {
393                         if (memcmp(secret, &(secret[hexkeybytes]),
394                                         keychecksumsize) != 0) {
395                                 secret[0] = 0;
396                                 return (0);
397                         }
398                 } else {
399                         MD5_CTX md5_ctx;
400                         char md5hexbuf[MD5HEXSIZE + 1] = {0};
401                         uint8_t digest[MD5HEXSIZE/2];

403                         MD5Init(&md5_ctx);
404                         MD5Update(&md5_ctx, (unsigned char *)secret,
405                                         hexkeybytes);
406                         MD5Final(digest, &md5_ctx);

408                         /* convert md5 binary digest to hex */
409                         (void) bin2hex(MD5HEXSIZE/2, digest, md5hexbuf);

411                         /* does the digest match the appended one? */
412                         if (memcmp(&(secret[hexkeybytes]),
413                                         md5hexbuf, MD5HEXSIZE) != 0) {
414                                 secret[0] = 0;
415                                 return (0);
416                         }
417                 }

419         secret[hexkeybytes] = '\0';

421         return (1);
446 #if 0
447 /* EXPORT DELETE END */
448         return (0);
449 /* EXPORT DELETE START */
450 #endif
451 /* EXPORT DELETE END */
422 }
```

_____unchanged_portion_omitted_

```
**********************************************************
    2144 Thu Jul 11 01:29:29 2013
new/usr/src/lib/libsasl/Makefile
first pass
**********************************************************
#
# 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 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
#

include ../Makefile.lib

HDRS=           sasl.h          saslplug.h      saslutil.h      prop.h
HDRDIR=         include
ROOTHDRDIR=     $(ROOT)/usr/include/sasl

SUBDIRS =       $(MACH)
$(BUILD64)SUBDIRS += $(MACH64)

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

LIBRARY=        libsasl.a
POFILE=         $(LIBRARY:.a=.po)
MSGFILES=       lib/canonusr.c lib/checkpw.c lib/client.c lib/common.c \
                        lib/external.c lib/server.c lib/seterror.c \
                        plugin/plugin_common.c \
                        $(SRC)/lib/sasl_plugins/cram/cram.c \
                        $(SRC)/lib/sasl_plugins/digestmd5/digestmd5.c \
                        $(SRC)/lib/sasl_plugins/gssapi/gssapi.c \
                        $(SRC)/lib/sasl_plugins/login/login.c \
                        $(SRC)/lib/sasl_plugins/plain/plain.c

.KEEP_STATE:

all clean clobber install lint: $(SUBDIRS)

all install: THIRDPARTYLICENSE

install_h:      $(ROOTHDRS)
```



```
check:  $(CHECKHDRS)

$(POFILE): $(MSGFILES)
        $(BUILDPO.msgfiles)

_msg: $(MSGDOMAINPOFILE)

include $(SRC)/Makefile.msg.targ

$(SUBDIRS): FRC
        @cd $@; pwd; $(MAKE) $(TARGET)

FRC:

THIRDPARTYLICENSE: LICENSE.txt
        $(SED) -n '/Carnegie Mellon/,$$p' LICENSE.txt > $@

CLOBBERFILES += THIRDPARTYLICENSE

# EXPORT DELETE START
# CRYPT DELETE START
# Special target to clean up the source tree for export distribution
# Warning: This target changes the source tree
EXPORT_SRC:
        $(RM) Makefile+ \
                lib/client.c+ \
                lib/server.c+ \
                lib/common.c+ \
                lib/saslint.h+ \
                include/plugin_common.h+

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < lib/client.c > lib/client.c+
        $(MV) lib/client.c+ lib/client.c

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < lib/server.c > lib/server.c+
        $(MV) lib/server.c+ lib/server.c

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < lib/common.c > lib/common.c+
        $(MV) lib/common.c+ lib/common.c

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < lib/saslint.h > lib/saslint.h+
        $(MV) lib/saslint.h+ lib/saslint.h

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < include/plugin_common.h > include/plugin_common.h+
        $(MV) include/plugin_common.h+ include/plugin_common.h

        $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile

        $(CHMOD) 444 Makefile \
                lib/client.c \
                lib/server.c \
                lib/common.c \
                lib/saslint.h \
                include/plugin_common.h

CRYPT_SRC:
        $(RM) Makefile+ lib/common.c+

        $(SED) -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
```

```
128                 < lib/common.c | $(SED) -e "/EXPORT DELETE/d" \
129                 > lib/common.c+
130         $(MV) lib/common.c+ lib/common.c

132         $(SED) -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
133                 < lib/client.c | $(SED) -e "/EXPORT DELETE/d" \
134                 > lib/client.c+
135         $(MV) lib/client.c+ lib/client.c

137         $(SED) -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
138                 < lib/server.c | $(SED) -e "/EXPORT DELETE/d" \
139                 > lib/server.c+
140         $(MV) lib/server.c+ lib/server.c

142         $(SED) -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
143                 < lib/saslint.h | $(SED) -e "/EXPORT DELETE/d" \
144                 > lib/saslint.h+
145         $(MV) lib/saslint.h+ lib/saslint.h

147         $(SED) -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
148                 < include/plugin_common.h | $(SED) -e "/EXPORT DELETE/d" \
149                 > include/plugin_common.h+
150         $(MV) include/plugin_common.h+ include/plugin_common.h

152         $(SED) -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d" \
153                 < Makefile | $(SED) -e "/^# EXPORT DELETE/d" > Makefile+
154         $(MV) Makefile+ Makefile

156         $(CHMOD) 444 Makefile \
157                 lib/client.c \
158                 lib/server.c \
159                 lib/common.c \
160                 lib/saslint.h \
161                 include/plugin_common.h

163 # CRYPT DELETE END
164 # EXPORT DELETE END
```

```
 81 include ../Makefile.targ

 83 .PARALLEL: $(SUBDIRS)
```

```
**********************************************************
    8199 Thu Jul 11 01:29:30 2013
new/usr/src/lib/libsasl/include/plugin_common.h
first pass
**********************************************************
______unchanged_portion_omitted_
```
```
151 #endif

153 int _plug_ipfromstring(const sasl_utils_t *utils, const char *addr,
154                         struct sockaddr *out, socklen_t outlen);
155 int _plug_iovec_to_buf(const sasl_utils_t *utils, const struct iovec *vec,
156                         unsigned numiov, buffer_info_t **output);
157 int _plug_buf_alloc(const sasl_utils_t *utils, char **rwbuf,
158                     unsigned *curlen, unsigned newlen);
159 int _plug_strdup(const sasl_utils_t * utils, const char *in,
160                  char **out, int *outlen);
161 void _plug_free_string(const sasl_utils_t *utils, char **str);
162 void _plug_free_secret(const sasl_utils_t *utils, sasl_secret_t **secret);

164 #define _plug_get_userid(utils, result, prompt_need) \
165         _plug_get_simple(utils, SASL_CB_USER, 0, result, prompt_need)
166 #define _plug_get_authid(utils, result, prompt_need) \
167         _plug_get_simple(utils, SASL_CB_AUTHNAME, 1, result, prompt_need)
168 int _plug_get_simple(const sasl_utils_t *utils, unsigned int id, int required,
169                      const char **result, sasl_interact_t **prompt_need);

171 int _plug_get_password(const sasl_utils_t *utils, sasl_secret_t **secret,
172                        unsigned int *iscopy, sasl_interact_t **prompt_need);

174 int _plug_challenge_prompt(const sasl_utils_t *utils, unsigned int id,
175                            const char *challenge, const char *promptstr,
176                            const char **result, sasl_interact_t **prompt_need);

178 int _plug_get_realm(const sasl_utils_t *utils, const char **availrealms,
179                     const char **realm, sasl_interact_t **prompt_need);

181 int _plug_make_prompts(const sasl_utils_t *utils,
182 #ifdef _INTEGRATED_SOLARIS_
183                        void **h,
184 #endif /* _INTEGRATED_SOLARIS_ */
185                        sasl_interact_t **prompts_res,
186                        const char *user_prompt, const char *user_def,
187                        const char *auth_prompt, const char *auth_def,
188                        const char *pass_prompt, const char *pass_def,
189                        const char *echo_chal,
190                        const char *echo_prompt, const char *echo_def,
191                        const char *realm_chal,
192                        const char *realm_prompt, const char *realm_def);

194 int _plug_decode(const sasl_utils_t *utils,
195                  void *context,
196                  const char *input, unsigned inputlen,
197                  char **output, unsigned *outputsize, unsigned *outputlen,
198                  int (*decode_pkt)(void *context,
199                                    const char **input, unsigned *inputlen,
200                                    char **output, unsigned *outputlen));

202 int _plug_parseuser(const sasl_utils_t *utils,
203                     char **user, char **realm, const char *user_realm,
204                     const char *serverFQDN, const char *input);

206 #ifdef _INTEGRATED_SOLARIS_
207 /* EXPORT DELETE START */
208 /* CRYPT DELETE START */
207 typedef void reg_sun_t(void *);
```



```
209 #define REG_PLUG( X, Y ) { \
210         reg_sun_t *func = NULL; \
211         unsigned int l; \
212         utils->getopt(utils->getopt_context, X, "reg_sun_plug", \
213                 (const char **)&func, &l); \
214         if (func != NULL && l == 0) \
215                 (*func)(Y); \
216 }
219 /* CRYPT DELETE END */
220 /* EXPORT DELETE END */

218 int use_locale(const char *lang_list, int is_client);
219 const char *convert_prompt(const sasl_utils_t *utils, void **h, const char *s);
220 char *local_to_utf(const sasl_utils_t *utils, const char *s);
221 #endif /* _INTEGRATED_SOLARIS_ */
222 #endif /* _PLUGIN_COMMON_H_ */
```

```
**********************************************************
   34283 Thu Jul 11 01:29:30 2013
new/usr/src/lib/libsasl/lib/client.c
first pass
**********************************************************
______unchanged_portion_omitted_

 194 int _sasl_client_add_plugin(void *ctx,
 195                            const char *plugname,
 196                            sasl_client_plug_init_t *entry_point)
 197 {
 198   cmech_list_t *cmechlist;
 199 #ifdef _INTEGRATED_SOLARIS_
 200   _sasl_global_context_t *gctx = ctx == NULL ? _sasl_gbl_ctx() : ctx;
 201   /* EXPORT DELETE START */
 202   /* CRYPT DELETE START */
 201   int sun_reg;
 204   /* CRYPT DELETE END */
 205   /* EXPORT DELETE END */
 202 #endif /* _INTEGRATED_SOLARIS_ */
 203   int i;
 204   cmechanism_t *m;
 205 #endif /* _SUN_SDK_ */
 206   int plugcount;
 207   sasl_client_plug_t *pluglist;
 208   cmechanism_t *mech;
 209   int result;
 210   int version;
 211   int lupe;

 213   if(!plugname || !entry_point) return SASL_BADPARAM;
 214
 215 #ifdef _SUN_SDK_
 216   cmechlist = gctx->cmechlist;

 218   if (cmechlist == NULL) return SASL_BADPARAM;

 220   /* Check to see if this plugin has already been registered */
 221   m = cmechlist->mech_list;
 222   for (i = 0; i < cmechlist->mech_length; i++) {
 223     if (strcmp(plugname, m->plugname) == 0) {
 224       return SASL_OK;
 225     }
 226     m = m->next;
 227   }

 229   result = LOCK_MUTEX(&client_plug_mutex);
 230   if (result != SASL_OK)
 231       return result;

 233 #endif /* _SUN_SDK_ */

 235   result = entry_point(cmechlist->utils, SASL_CLIENT_PLUG_VERSION, &version,
 236                        &pluglist, &plugcount);

 242   /* EXPORT DELETE START */
 243   /* CRYPT DELETE START */
 238 #ifdef _INTEGRATED_SOLARIS_
 239   sun_reg = _is_sun_reg(pluglist);
 240 #endif /* _INTEGRATED_SOLARIS_ */
 247   /* CRYPT DELETE END */
 248   /* EXPORT DELETE END */
 241   if (result != SASL_OK)
 242   {
 243 #ifdef _SUN_SDK_
 244     UNLOCK_MUTEX(&client_plug_mutex);
```



```
 245     __sasl_log(gctx, gctx->client_global_callbacks.callbacks, SASL_LOG_WARN,
 246                "entry_point failed in sasl_client_add_plugin for %s",
 247                plugname);
 248 #else
 249     _sasl_log(NULL, SASL_LOG_WARN,
 250                "entry_point failed in sasl_client_add_plugin for %s",
 251                plugname);
 252 #endif /* _SUN_SDK_ */
 253     return result;
 254   }

 256   if (version != SASL_CLIENT_PLUG_VERSION)
 257   {
 258 #ifdef _SUN_SDK_
 259     UNLOCK_MUTEX(&client_plug_mutex);
 260     __sasl_log(gctx, gctx->client_global_callbacks.callbacks, SASL_LOG_WARN,
 261                "version conflict in sasl_client_add_plugin for %s", plugname);
 262 #else
 263     _sasl_log(NULL, SASL_LOG_WARN,
 264                "version conflict in sasl_client_add_plugin for %s", plugname);
 265 #endif /* _SUN_SDK_ */
 266     return SASL_BADVERS;
 267   }

 269 #ifdef _SUN_SDK_
 270     /* Check plugins to make sure mech_name is non-NULL */
 271     for (lupe=0;lupe < plugcount ;lupe++) {
 272         if (pluglist[lupe].mech_name == NULL)
 273             break;
 274     }
 275     if (lupe < plugcount) {
 276         UNLOCK_MUTEX(&client_plug_mutex);
 277         __sasl_log(gctx, gctx->client_global_callbacks.callbacks,
 278                 SASL_LOG_ERR, "invalid client plugin %s", plugname);
 279         return SASL_BADPROT;
 280     }
 281 #endif /* _SUN_SDK_ */

 283   for (lupe=0;lupe< plugcount ;lupe++)
 284     {
 285       mech = sasl_ALLOC(sizeof(cmechanism_t));
 286 #ifdef _SUN_SDK_
 287       if (! mech) {
 288         UNLOCK_MUTEX(&client_plug_mutex);
 289         return SASL_NOMEM;
 290       }
 291       mech->glob_context = pluglist->glob_context;
 292 #else
 293       if (! mech) return SASL_NOMEM;
 294 #endif /* _SUN_SDK_ */

 296       mech->plug=pluglist++;
 297       if(_sasl_strdup(plugname, &mech->plugname, NULL) != SASL_OK) {
 298 #ifdef _SUN_SDK_
 299         UNLOCK_MUTEX(&client_plug_mutex);
 300 #endif /* _SUN_SDK_ */
 301         sasl_FREE(mech);
 302         return SASL_NOMEM;
 303       }
 312       /* EXPORT DELETE START */
 313       /* CRYPT DELETE START */
 304 #ifdef _INTEGRATED_SOLARIS_
 305       mech->sun_reg = sun_reg;
 306 #endif /* _INTEGRATED_SOLARIS_ */
 317      /* CRYPT DELETE END */
 318      /* EXPORT DELETE END */
```

```
 307       mech->version = version;
 308       mech->next = cmechlist->mech_list;
 309       cmechlist->mech_list = mech;
 310       cmechlist->mech_length++;
 311     }
 312 #ifdef _SUN_SDK_
 313     UNLOCK_MUTEX(&client_plug_mutex);
 314 #endif /* _SUN_SDK_ */

 316   return SASL_OK;
 317 }
_____unchanged_portion_omitted_

 711 /* select a mechanism for a connection
 712  *  mechlist      -- mechanisms server has available (punctuation ignored)
 713  *  secret        -- optional secret from previous session
 714  * output:
 715  *  prompt_need   -- on SASL_INTERACT, list of prompts needed to continue
 716  *  clientout     -- the initial client response to send to the server
 717  *  mech          -- set to mechanism name
 718  *
 719  * Returns:
 720  *  SASL_OK       -- success
 721  *  SASL_NOMEM    -- not enough memory
 722  *  SASL_NOMECH   -- no mechanism meets requested properties
 723  *  SASL_INTERACT -- user interaction needed to fill in prompt_need list
 724  */

 726 /* xxx confirm this with rfc 2222
 727  * SASL mechanism allowable characters are "AZaz-_"
 728  * seperators can be any other characters and of any length
 729  * even variable lengths between
 730  *
 731  * Apps should be encouraged to simply use space or comma space
 732  * though
 733  */
 734 int sasl_client_start(sasl_conn_t *conn,
 735                       const char *mechlist,
 736                       sasl_interact_t **prompt_need,
 737                       const char **clientout,
 738                       unsigned *clientoutlen,
 739                       const char **mech)
 740 {
 741     sasl_client_conn_t *c_conn= (sasl_client_conn_t *) conn;
 742     char name[SASL_MECHNAMEMAX + 1];
 743     cmechanism_t *m=NULL,*bestm=NULL;
 744     size_t pos=0,place;
 745     size_t list_len;
 746     sasl_ssf_t bestssf = 0, minssf = 0;
 747     int result;
 748 #ifdef _SUN_SDK_
 749     _sasl_global_context_t *gctx = (conn == NULL) ?
 750                  _sasl_gbl_ctx() : conn->gctx;
 751     cmech_list_t *cmechlist;

 753     if(gctx->sasl_client_active==0) return SASL_NOTINIT;
 754     cmechlist = gctx->cmechlist;
 755 #else
 756     if(_sasl_client_active==0) return SASL_NOTINIT;
 757 #endif /* _SUN_SDK_ */

 759     if (!conn) return SASL_BADPARAM;

 761     /* verify parameters */
 762     if (mechlist == NULL)
 763         PARAMERROR(conn);
```



```
 765     /* if prompt_need != NULL we've already been here
 766        and just need to do the continue step again */

 768     /* do a step */
 769     /* FIXME: Hopefully they only give us our own prompt_need back */
 770     if (prompt_need && *prompt_need != NULL) {
 771         goto dostep;
 772     }

 774 #ifdef _SUN_SDK_
 775     if (c_conn->mech != NULL) {
 776         if (c_conn->mech->plug->mech_dispose != NULL) {
 777             c_conn->mech->plug->mech_dispose(conn->context,
 778                 c_conn->cparams->utils);
 779             c_conn->mech = NULL;
 780         }
 781     }
 782     memset(&conn->oparams, 0, sizeof(sasl_out_params_t));

 784     (void) _load_client_plugins(gctx);
 785 #endif /* _SUN_SDK_ */

 787     if(conn->props.min_ssf < conn->external.ssf) {
 788         minssf = 0;
 789     } else {
 790         minssf = conn->props.min_ssf - conn->external.ssf;
 791     }

 793     /* parse mechlist */
 794     list_len = strlen(mechlist);

 796     while (pos<list_len)
 797     {
 798         place=0;
 799         while ((pos<list_len) && (isalnum((unsigned char)mechlist[pos])
 800                                   || mechlist[pos] == '_'
 801                                   || mechlist[pos] == '-')) {
 802             name[place]=mechlist[pos];
 803             pos++;
 804             place++;
 805             if (SASL_MECHNAMEMAX < place) {
 806                 place--;
 807                 while(pos<list_len && (isalnum((unsigned char)mechlist[pos])
 808                                        || mechlist[pos] == '_'
 809                                        || mechlist[pos] == '-'))
 810                     pos++;
 811             }
 812         }
 813         pos++;
 814         name[place]=0;

 816         if (! place) continue;

 818         /* foreach in server list */
 819         for (m = cmechlist->mech_list; m != NULL; m = m->next) {
 820             int myflags;
 821
 822             /* Is this the mechanism the server is suggesting? */
 823             if (strcasecmp(m->plug->mech_name, name))
 824                 continue; /* no */

 826             /* Do we have the prompts for it? */
 827             if (!have_prompts(conn, m->plug))
 828                 break;
```

```
            /* Is it strong enough? */
            if (minssf > m->plug->max_ssf)
                break;

            /* EXPORT DELETE START */
            /* CRYPT DELETE START */
#ifdef _INTEGRATED_SOLARIS_
            /* If not SUN supplied mech, it has no strength */
            if (minssf > 0 && !m->sun_reg)
                break;
#endif /* _INTEGRATED_SOLARIS_ */
            /* CRYPT DELETE END */
            /* EXPORT DELETE END */

            /* Does it meet our security properties? */
            myflags = conn->props.security_flags;

            /* if there's an external layer this is no longer plaintext */
            if ((conn->props.min_ssf <= conn->external.ssf) &&
                (conn->external.ssf > 1)) {
                myflags &= ~SASL_SEC_NOPLAINTEXT;
            }

            if (((myflags ^ m->plug->security_flags) & myflags) != 0) {
                break;
            }

            /* Can we meet it's features? */
            if ((m->plug->features & SASL_FEAT_NEEDSERVERFQDN)
                && !conn->serverFQDN) {
                break;
            }

            /* Can it meet our features? */
            if ((conn->flags & SASL_NEED_PROXY) &&
                !(m->plug->features & SASL_FEAT_ALLOWS_PROXY)) {
                break;
            }

#ifdef PREFER_MECH
            /* EXPORT DELETE START */
            /* CRYPT DELETE START */
#ifdef _INTEGRATED_SOLARIS_
            if (strcasecmp(m->plug->mech_name, PREFER_MECH) &&
                bestm && (m->sun_reg && m->plug->max_ssf <= bestssf) ||
                (m->plug->max_ssf == 0)) {
#else
            /* CRYPT DELETE END */
            /* EXPORT DELETE END */
            if (strcasecmp(m->plug->mech_name, PREFER_MECH) &&
                bestm && m->plug->max_ssf <= bestssf) {

                /* EXPORT DELETE START */
                /* CRYPT DELETE START */
#endif /* _INTEGRATED_SOLARIS_ */
                /* CRYPT DELETE END */
                /* EXPORT DELETE END */

                /* this mechanism isn't our favorite, and it's no better
                   than what we already have! */
                break;
            }
#else
            /* EXPORT DELETE START */
            /* CRYPT DELETE START */
#ifdef _INTEGRATED_SOLARIS_
```



```
            if (bestm && m->sun_reg && m->plug->max_ssf <= bestssf) {
#else
            /* CRYPT DELETE END */
            /* EXPORT DELETE END */

            if (bestm && m->plug->max_ssf <= bestssf) {
            /* EXPORT DELETE START */
            /* CRYPT DELETE START */
#endif /* _INTEGRATED_SOLARIS_ */
            /* CRYPT DELETE END */
            /* EXPORT DELETE END */

                /* this mechanism is no better than what we already have! */
                break;
            }
#endif

            /* compare security flags, only take new mechanism if it has
             * all the security flags of the previous one.
             *
             * From the mechanisms we ship with, this yields the order:
             *
             * SRP
             * GSSAPI + KERBEROS_V4
             * DIGEST + OTP
             * CRAM + EXTERNAL
             * PLAIN + LOGIN + ANONYMOUS
             *
             * This might be improved on by comparing the numeric value of
             * the bitwise-or'd security flags, which splits DIGEST/OTP,
             * CRAM/EXTERNAL, and PLAIN/LOGIN from ANONYMOUS, but then we
             * are depending on the numeric values of the flags (which may
             * change, and their ordering could be considered dumb luck.
             */

            if (bestm &&
                ((m->plug->security_flags ^ bestm->plug->security_flags) &
                 bestm->plug->security_flags)) {
                break;
            }

            if (mech) {
                *mech = m->plug->mech_name;
            }
            /* EXPORT DELETE START */
            /* CRYPT DELETE START */
#ifdef _INTEGRATED_SOLARIS_
            bestssf = m->sun_reg ? m->plug->max_ssf : 0;
#else
            /* CRYPT DELETE END */
            /* EXPORT DELETE END */
            bestssf = m->plug->max_ssf;
            /* EXPORT DELETE START */
            /* CRYPT DELETE START */
#endif /* _INTEGRATED_SOLARIS_ */
            /* CRYPT DELETE END */
            /* EXPORT DELETE END */
            bestm = m;
            break;
        }
    }

    if (bestm == NULL) {
#ifdef _INTEGRATED_SOLARIS_
        sasl_seterror(conn, 0, gettext("No worthy mechs found"));
#else
```

```
 933         sasl_seterror(conn, 0, "No worthy mechs found");
 934 #endif /* _INTEGRATED_SOLARIS_ */
 935         result = SASL_NOMECH;
 936         goto done;
 937     }

 939     /* make (the rest of) cparams */
 940     c_conn->cparams->service = conn->service;
 941     c_conn->cparams->servicelen = strlen(conn->service);
 942
 943     c_conn->cparams->serverFQDN = conn->serverFQDN;
 944     c_conn->cparams->slen = strlen(conn->serverFQDN);

 946     c_conn->cparams->clientFQDN = c_conn->clientFQDN;
 947     c_conn->cparams->clen = strlen(c_conn->clientFQDN);

 949     c_conn->cparams->external_ssf = conn->external.ssf;
 950     c_conn->cparams->props = conn->props;
 992     /* EXPORT DELETE START */
 993     /* CRYPT DELETE START */
 951 #ifdef _INTEGRATED_SOLARIS_
 952     if (!bestm->sun_reg) {
 953         c_conn->cparams->props.min_ssf = 0;
 954         c_conn->cparams->props.max_ssf = 0;
 955     }
 956     c_conn->base.sun_reg = bestm->sun_reg;
 957 #endif /* _INTEGRATED_SOLARIS_ */
1001     /* CRYPT DELETE END */
1002     /* EXPORT DELETE END */
 958     c_conn->mech = bestm;

 960     /* init that plugin */
 961 #ifdef _SUN_SDK_
 962     result = c_conn->mech->plug->mech_new(c_conn->mech->glob_context,
 963 #else
 964     result = c_conn->mech->plug->mech_new(c_conn->mech->plug->glob_context,
 965 #endif /* _SUN_SDK_ */
 966                                           c_conn->cparams,
 967                                           &(conn->context));
 968     if(result != SASL_OK) goto done;

 970     /* do a step -- but only if we can do a client-send-first */
 971  dostep:
 972     if(clientout) {
 973         if(c_conn->mech->plug->features & SASL_FEAT_SERVER_FIRST) {
 974             *clientout = NULL;
 975             *clientoutlen = 0;
 976             result = SASL_CONTINUE;
 977         } else {
 978             result = sasl_client_step(conn, NULL, 0, prompt_need,
 979                                       clientout, clientoutlen);
 980         }
 981     }
 982     else
 983         result = SASL_CONTINUE;

 985  done:
 986     RETURN(conn, result);
 987 }
______unchanged_portion_omitted_


1098 int _sasl_client_listmech(sasl_conn_t *conn,
1099                           const char *prefix,
1100                           const char *sep,
1101                           const char *suffix,
```



```
1102                           const char **result,
1103                           unsigned *plen,
1104                           int *pcount)
1105 {
1106     cmechanism_t *m=NULL;
1107     sasl_ssf_t minssf = 0;
1108     int ret;
1109     unsigned int resultlen;
1110     int flag;
1111     const char *mysep;
1112 #ifdef _SUN_SDK_
1113     _sasl_global_context_t *gctx = conn == NULL ? _sasl_gbl_ctx() : conn->gctx;
1114     cmech_list_t *cmechlist;

1116     if(gctx->sasl_client_active==0) return SASL_NOTINIT;
1117     cmechlist = gctx->cmechlist;
1118 #else
1119     if(_sasl_client_active == 0) return SASL_NOTINIT;
1120 #endif /* _SUN_SDK_ */
1121     if (!conn) return SASL_BADPARAM;
1122     if(conn->type != SASL_CONN_CLIENT) PARAMERROR(conn);
1123
1124     if (! result)
1125         PARAMERROR(conn);
1126
1127 #ifdef _SUN_SDK_
1128      (void) _load_client_plugins(gctx);
1129 #endif /* _SUN_SDK_ */

1131     if (plen != NULL)
1132         *plen = 0;
1133     if (pcount != NULL)
1134         *pcount = 0;

1136     if (sep) {
1137         mysep = sep;
1138     } else {
1139         mysep = " ";
1140     }

1142     if(conn->props.min_ssf < conn->external.ssf) {
1143         minssf = 0;
1144     } else {
1145         minssf = conn->props.min_ssf - conn->external.ssf;
1146     }

1148     if (! cmechlist || cmechlist->mech_length <= 0)
1149         INTERROR(conn, SASL_NOMECH);

1151     resultlen = (prefix ? strlen(prefix) : 0)
1152         + (strlen(mysep) * (cmechlist->mech_length - 1))
1153 #ifdef _SUN_SDK_
1154         + mech_names_len(gctx)
1155 #else
1156         + mech_names_len()
1157 #endif /* _SUN_SDK_ */
1158         + (suffix ? strlen(suffix) : 0)
1159         + 1;
1160     ret = _buf_alloc(&conn->mechlist_buf,
1161                      &conn->mechlist_buf_len, resultlen);
1162     if(ret != SASL_OK) MEMERROR(conn);

1164     if (prefix)
1165         strcpy (conn->mechlist_buf,prefix);
1166     else
1167         *(conn->mechlist_buf) = '\0';
```

```
1169     flag = 0;
1170     for (m = cmechlist->mech_list; m != NULL; m = m->next) {
1171             /* do we have the prompts for it? */
1172             if (!have_prompts(conn, m->plug))
1173                 continue;

1175             /* is it strong enough? */
1176             if (minssf > m->plug->max_ssf)
1177                 continue;

1224             /* EXPORT DELETE START */
1225             /* CRYPT DELETE START */
1179 #ifdef _INTEGRATED_SOLARIS_
1180             /* If not SUN supplied mech, it has no strength */
1181             if (minssf > 0 && !m->sun_reg)
1182                 continue;
1183 #endif /* _INTEGRATED_SOLARIS_ */
1231             /* CRYPT DELETE END */
1232             /* EXPORT DELETE END */

1185             /* does it meet our security properties? */
1186             if (((conn->props.security_flags ^ m->plug->security_flags)
1187                  & conn->props.security_flags) != 0) {
1188                 continue;
1189             }

1191             /* Can we meet it's features? */
1192             if ((m->plug->features & SASL_FEAT_NEEDSERVERFQDN)
1193                 && !conn->serverFQDN) {
1194                 continue;
1195             }

1197             /* Can it meet our features? */
1198             if ((conn->flags & SASL_NEED_PROXY) &&
1199                 !(m->plug->features & SASL_FEAT_ALLOWS_PROXY)) {
1200                 break;
1201             }

1203             /* Okay, we like it, add it to the list! */

1205             if (pcount != NULL)
1206                 (*pcount)++;

1208             /* print seperator */
1209             if (flag) {
1210                 strcat(conn->mechlist_buf, mysep);
1211             } else {
1212                 flag = 1;
1213             }
1214
1215             /* now print the mechanism name */
1216             strcat(conn->mechlist_buf, m->plug->mech_name);
1217     }
1218
1219   if (suffix)
1220       strcat(conn->mechlist_buf,suffix);

1222   if (plen!=NULL)
1223       *plen=strlen(conn->mechlist_buf);

1225   *result = conn->mechlist_buf;

1227   return SASL_OK;
1228 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   75422 Thu Jul 11 01:29:31 2013
new/usr/src/lib/libsasl/lib/common.c
pass 2
first pass
**********************************************************
______unchanged_portion_omitted_
```

```
 296 /* security-encode an iovec */
 297 /* output is only valid until next call to sasl_encode or sasl_encodev */
 298 int sasl_encodev(sasl_conn_t *conn,
 299                  const struct iovec *invec, unsigned numiov,
 300                  const char **output, unsigned *outputlen)
 301 {
 302 #ifdef _SUN_SDK_
 303     int result = SASL_FAIL;
 304 #else
 305     int result;
 306 #endif /* _SUN_SDK_ */
 307     unsigned i;
 308     size_t total_size = 0;

 310     /* EXPORT DELETE START */
 310     if (!conn) return SASL_BADPARAM;
 311     if (! invec || ! output || ! outputlen || numiov < 1)
 312         PARAMERROR(conn);

 314     if(!conn->props.maxbufsize) {
 315 #ifdef _SUN_SDK_
 316         _sasl_log(conn, SASL_LOG_ERR,
 317                   "called sasl_encode[v] with application that does not support
 318 #else
 319         sasl_seterror(conn, 0,
 320                       "called sasl_encode[v] with application that does not supp
 321 #endif /* _SUN_SDK_ */
 322         return SASL_TOOWEAK;
 323     }

 325     /* This might be better to check on a per-plugin basis, but I think
 326      * it's cleaner and more effective here.  It also encourages plugins
 327      * to be honest about what they accept */

 329     for(i=0; i<numiov;i++) {
 330 #ifdef _SUN_SDK_
 331         if (invec[i].iov_base == NULL)
 332             PARAMERROR(conn);
 333 #endif /* _SUN_SDK_ */
 334         total_size += invec[i].iov_len;
 335     }
 336     if(total_size > conn->oparams.maxoutbuf)
 337         PARAMERROR(conn);

 339     if(conn->oparams.encode == NULL)  {
 340 #ifdef _SUN_SDK_
 341         result = _iovec_to_buf(conn->gctx, invec, numiov, &conn->encode_buf);
 342 #else
 343         result = _iovec_to_buf(invec, numiov, &conn->encode_buf);
 344 #endif /* _SUN_SDK_ */
 345         if(result != SASL_OK) INTERROR(conn, result);
 346
 347         *output = conn->encode_buf->data;
 348         *outputlen = conn->encode_buf->curlen;

 351     /* CRYPT DELETE START */
 350 #ifdef _INTEGRATED_SOLARIS_
 351     } else if (!conn->sun_reg) {
```



```
 352             INTERROR(conn, SASL_FAIL);
 353 #endif /* _INTEGRATED_SOLARIS_ */
 356     /* CRYPT DELETE END */
 354     } else {
 355         result = conn->oparams.encode(conn->context, invec, numiov,
 356                                       output, outputlen);
 357     }
 361     /* EXPORT DELETE END */

 359     RETURN(conn, result);
 360 }
 361
 362 /* output is only valid until next call to sasl_decode */
 363 int sasl_decode(sasl_conn_t *conn,
 364                 const char *input, unsigned inputlen,
 365                 const char **output, unsigned *outputlen)
 366 {
 367     int result;
 372     /* EXPORT DELETE START */
 368 #ifdef _SUN_SDK_
 369     const _sasl_global_context_t *gctx;
 370 #endif /* _SUN_SDK_ */

 372     if(!conn) return SASL_BADPARAM;
 373     if(!input || !output || !outputlen)
 374         PARAMERROR(conn);

 376 #ifdef _SUN_SDK_
 377     gctx = conn->gctx;
 378 #endif /* _SUN_SDK_ */

 380     if(!conn->props.maxbufsize) {
 381 #ifdef _SUN_SDK_
 382         _sasl_log(conn, SASL_LOG_ERR,
 383                   "called sasl_decode with application that does not support sec
 384 #else
 385         sasl_seterror(conn, 0,
 386                       "called sasl_decode with application that does not support
 387 #endif /* _SUN_SDK_ */
 388         RETURN(conn, SASL_TOOWEAK);
 389     }

 391     if(conn->oparams.decode == NULL)
 392     {
 393         /* Since we know how long the output is maximally, we can
 394          * just allocate it to begin with, and never need another
 395          * allocation! */

 397         /* However, if they pass us more than they actually can take,
 398          * we cannot help them... */
 399         if(inputlen > conn->props.maxbufsize) {
 400 #ifdef _SUN_SDK_
 401             _sasl_log(conn, SASL_LOG_ERR,
 402                       "input too large for default sasl_decode");
 403 #else
 404             sasl_seterror(conn, 0,
 405                           "input too large for default sasl_decode");
 406 #endif /* _SUN_SDK_ */
 407             RETURN(conn,SASL_BUFOVER);
 408         }

 410         if(!conn->decode_buf)
 411             conn->decode_buf = sasl_ALLOC(conn->props.maxbufsize + 1);
 412         if(!conn->decode_buf)
 413             MEMERROR(conn);
 414
```

```
415         memcpy(conn->decode_buf, input, inputlen);
416         conn->decode_buf[inputlen] = '\0';
417         *output = conn->decode_buf;
418         *outputlen = inputlen;
419
420         return SASL_OK;
426     /* CRYPT DELETE START */
421 #ifdef _INTEGRATED_SOLARIS_
422     } else if (!conn->sun_reg) {
423             INTERROR(conn, SASL_FAIL);
424 #endif /* _INTEGRATED_SOLARIS_ */
431     /* CRYPT DELETE END */
425     } else {
426         result = conn->oparams.decode(conn->context, input, inputlen,
427                                       output, outputlen);

429         /* NULL an empty buffer (for misbehaved applications) */
430         if (*outputlen == 0) *output = NULL;

432         RETURN(conn, result);
433     }

442     /* EXPORT DELETE END */
435 #ifdef _SUN_SDK_
436     return SASL_FAIL;
437 #else
438     INTERROR(conn, SASL_FAIL);
439 #endif  /* _SUN_SDK_ */
440 }
```

unchanged_portion_omitted_

```
722 /* get property from SASL connection state
723  *  propnum       -- property number
724  *  pvalue        -- pointer to value
725  * returns:
726  *  SASL_OK       -- no error
727  *  SASL_NOTDONE  -- property not available yet
728  *  SASL_BADPARAM -- bad property number
729  */
730 int sasl_getprop(sasl_conn_t *conn, int propnum, const void **pvalue)
731 {
732   int result = SASL_OK;
733   sasl_getopt_t *getopt;
734   void *context;
735
736   if (! conn) return SASL_BADPARAM;
737   if (! pvalue) PARAMERROR(conn);

739   switch(propnum)
740   {
741   case SASL_SSF:
750     /* EXPORT DELETE START */
751     /* CRYPT DELETE START */
742 #ifdef _INTEGRATED_SOLARIS_
743       if (!conn->sun_reg)
744         conn->oparams.mech_ssf = 0;
745 #endif /* _INTEGRATED_SOLARIS_ */
756     /* CRYPT DELETE END */
757     /* EXPORT DELETE END */
746       *(sasl_ssf_t **)pvalue= &conn->oparams.mech_ssf;
747       break;
748   case SASL_MAXOUTBUF:
749       *(unsigned **)pvalue = &conn->oparams.maxoutbuf;
750       break;
751   case SASL_GETOPTCTX:
```



```
752       result = _sasl_getcallback(conn, SASL_CB_GETOPT, &getopt, &context);
753       if(result != SASL_OK) break;
754
755       *(void **)pvalue = context;
756       break;
757   case SASL_CALLBACK:
758       *(const sasl_callback_t **)pvalue = conn->callbacks;
759       break;
760   case SASL_IPLOCALPORT:
761       if(conn->got_ip_local)
762           *(const char **)pvalue = conn->iplocalport;
763       else {
764           *(const char **)pvalue = NULL;
765           result = SASL_NOTDONE;
766       }
767       break;
768   case SASL_IPREMOTEPORT:
769       if(conn->got_ip_remote)
770           *(const char **)pvalue = conn->ipremoteport;
771       else {
772           *(const char **)pvalue = NULL;
773           result = SASL_NOTDONE;
774       }
775       break;
776   case SASL_USERNAME:
777       if(! conn->oparams.user)
778           result = SASL_NOTDONE;
779       else
780           *((const char **)pvalue) = conn->oparams.user;
781       break;
782   case SASL_AUTHUSER:
783       if(! conn->oparams.authid)
784           result = SASL_NOTDONE;
785       else
786           *((const char **)pvalue) = conn->oparams.authid;
787       break;
788   case SASL_SERVERFQDN:
789       *((const char **)pvalue) = conn->serverFQDN;
790       break;
791   case SASL_DEFUSERREALM:
792       if(conn->type != SASL_CONN_SERVER) result = SASL_BADPROT;
793       else
794           *((const char **)pvalue) = ((sasl_server_conn_t *)conn)->user_realm;
795       break;
796   case SASL_SERVICE:
797       *((const char **)pvalue) = conn->service;
798       break;
799   case SASL_AUTHSOURCE: /* name of plugin (not name of mech) */
800       if(conn->type == SASL_CONN_CLIENT) {
801           if(!((sasl_client_conn_t *)conn)->mech) {
802               result = SASL_NOTDONE;
803               break;
804           }
805           *((const char **)pvalue) =
806               ((sasl_client_conn_t *)conn)->mech->plugname;
807       } else if (conn->type == SASL_CONN_SERVER) {
808           if(!((sasl_server_conn_t *)conn)->mech) {
809               result = SASL_NOTDONE;
810               break;
811           }
812           *((const char **)pvalue) =
813               ((sasl_server_conn_t *)conn)->mech->plugname;
814       } else {
815           result = SASL_BADPARAM;
816       }
817       break;
```

```
 818   case SASL_MECHNAME: /* name of mech */
 819       if(conn->type == SASL_CONN_CLIENT) {
 820           if(!((sasl_client_conn_t *)conn)->mech) {
 821               result = SASL_NOTDONE;
 822               break;
 823           }
 824           *((const char **)pvalue) =
 825               ((sasl_client_conn_t *)conn)->mech->plug->mech_name;
 826       } else if (conn->type == SASL_CONN_SERVER) {
 827           if(!((sasl_server_conn_t *)conn)->mech) {
 828               result = SASL_NOTDONE;
 829               break;
 830           }
 831           *((const char **)pvalue) =
 832               ((sasl_server_conn_t *)conn)->mech->plug->mech_name;
 833       } else {
 834           result = SASL_BADPARAM;
 835       }
 836
 837       if(!(*pvalue) && result == SASL_OK) result = SASL_NOTDONE;
 838       break;
 839   case SASL_PLUGERR:
 840       *((const char **)pvalue) = conn->error_buf;
 841       break;
 842   case SASL_SSF_EXTERNAL:
 843       *((const sasl_ssf_t **)pvalue) = &conn->external.ssf;
 844       break;
 845   case SASL_AUTH_EXTERNAL:
 846       *((const char **)pvalue) = conn->external.auth_id;
 847       break;
 848   case SASL_SEC_PROPS:
 849       *((const sasl_security_properties_t **)pvalue) = &conn->props;
 850       break;
 851   default:
 852       result = SASL_BADPARAM;
 853   }

 855   if(result == SASL_BADPARAM) {
 856       PARAMERROR(conn);
 857   } else if(result == SASL_NOTDONE) {
 858 #ifdef _SUN_SDK_
 859       _sasl_log(conn, SASL_LOG_NONE,
 860                 "Information that was requested is not yet available.");
 861 #else
 862       sasl_seterror(conn, SASL_NOLOG,
 863                     "Information that was requested is not yet available.");
 864 #endif /* _SUN_SDK_ */
 865       RETURN(conn, result);
 866   } else if(result != SASL_OK) {
 867       INTERROR(conn, result);
 868   } else
 869       RETURN(conn, result);
 870 #ifdef _SUN_SDK_
 871   return SASL_OK;
 872 #endif /* _SUN_SDK_ */
 873 }
```

unchanged_portion_omitted_

```
1307 /* EXPORT DELETE START */
1308 /* CRYPT DELETE START */
1295 #ifdef _INTEGRATED_SOLARIS_
1296 DEFINE_STATIC_MUTEX(reg_mutex);
1297 typedef struct reg_list {
1298         struct reg_list *next;
1299         void *mech;
1300 } reg_list_t;
```

unchanged_portion_omitted_


```
1341 #endif /* _INTEGRATED_SOLARIS_ */
1356 /* CRYPT DELETE END */
1357 /* EXPORT DELETE END */

1343 /* Note that this needs the global callbacks, so if you don't give getcallbacks
1344  * a sasl_conn_t, you're going to need to pass it yourself (or else we couldn't
1345  * have client and server at the same time */
1346 static int _sasl_global_getopt(void *context,
1347                                const char *plugin_name,
1348                                const char *option,
1349                                const char ** result,
1350                                unsigned *len)
1351 {
1352   const sasl_global_callbacks_t * global_callbacks;
1353   const sasl_callback_t *callback;
1354 #ifdef _SUN_SDK_
1355   _sasl_global_context_t *gctx;
1356 #endif /* _SUN_SDK_ */

1358   global_callbacks = (const sasl_global_callbacks_t *) context;

1360 #ifdef _SUN_SDK_
1377   /* EXPORT DELETE START */
1378   /* CRYPT DELETE START */
1361 #ifdef _INTEGRATED_SOLARIS_
1362   if (strcmp("reg_sun_plug", option) == 0) {
1363         *result = (const char *)_register_plugin;
1364         *len = 0;
1365         return (SASL_OK);
1366   }
1367 #endif /* _INTEGRATED_SOLARIS_ */
1386   /* CRYPT DELETE END */
1387   /* EXPORT DELETE END */

1369   if (global_callbacks)
1370     gctx = global_callbacks->gctx;
1371   else
1372     gctx = _sasl_gbl_ctx();
1373 #endif /* _SUN_SDK_ */

1375   if (global_callbacks && global_callbacks->callbacks) {
1376       for (callback = global_callbacks->callbacks;
1377            callback->id != SASL_CB_LIST_END;
1378            callback++) {
1379         if (callback->id == SASL_CB_GETOPT) {
1380           if (!callback->proc) return SASL_FAIL;
1381           if (((sasl_getopt_t *)(callback->proc))(callback->context,
1382                                                   plugin_name,
1383                                                   option,
1384                                                   result,
1385                                                   len)
1386               == SASL_OK)
1387             return SASL_OK;
1388         }
1389       }
1390   }
1391
1392   /* look it up in our configuration file */
1393 #ifdef _SUN_SDK_
1394   *result = sasl_config_getstring(gctx, option, NULL);
1395 #else
1396   *result = sasl_config_getstring(option, NULL);
1397 #endif /* _SUN_SDK_ */
1398   if (*result != NULL) {
1399       if (len) { *len = strlen(*result); }
1400       return SASL_OK;
```

```
1401   }

1403   return SASL_FAIL;
1404 }
______unchanged_portion_omitted_

2699 /* EXPORT DELETE START */
2700 /* CRYPT DELETE START */
2679 #ifdef _INTEGRATED_SOLARIS_
2680 #pragma fini(sasl_fini)
2681 int
2682 sasl_fini(void)
2683 {
2684     reg_list_t *next;

2686     while (reg_list_base != NULL) {
2687         next = reg_list_base->next;
2688         free(reg_list_base);
2689         reg_list_base = next;
2690     }
2691     return (0);
2692 }
2693 #endif /* _INTEGRATED_SOLARIS_ */
2716 /* CRYPT DELETE END */
2717 /* EXPORT DELETE END */

2695 #endif /* _SUN_SDK_ */

2697 #ifndef WIN32
2698 static int
2699 _sasl_getpath(void *context __attribute__((unused)),
2700               const char **path)
2701 {
2702   if (! path)
2703     return SASL_BADPARAM;

2705 #ifdef _SUN_SDK_
2706 /* SASL_PATH is not allowed for SUN SDK */
2707 #else
2708   *path = getenv(SASL_PATH_ENV_VAR);
2709   if (! *path)
2710 #endif /* _SUN_SDK_ */
2711     *path = PLUGINDIR;

2713   return SASL_OK;
2714 }
______unchanged_portion_omitted_
```

```
**********************************************************
   25453 Thu Jul 11 01:29:32 2013
new/usr/src/lib/libsasl/lib/saslint.h
first pass
**********************************************************
______unchanged_portion_omitted_

 165 enum Sasl_conn_type { SASL_CONN_UNKNOWN = 0,
 166                       SASL_CONN_SERVER = 1,
 167                       SASL_CONN_CLIENT = 2 };

 169 struct sasl_conn {
 170   enum Sasl_conn_type type;

 172   void (*destroy_conn)(sasl_conn_t *); /* destroy function */

 174   char *service;

 176   unsigned int flags;  /* flags passed to sasl_*_new */

 178   /* IP information.  A buffer of size 52 is adequate for this in its
 179      longest format (see sasl.h) */
 180   int got_ip_local, got_ip_remote;
 181   char iplocalport[NI_MAXHOST + NI_MAXSERV];
 182   char ipremoteport[NI_MAXHOST + NI_MAXSERV];

 184   void *context;
 185   sasl_out_params_t oparams;

 187   sasl_security_properties_t props;
 188   _sasl_external_properties_t external;

 190 #ifndef _SUN_SDK_
 191   sasl_secret_t *secret;
 192 #endif /* !_SUN_SDK_ */

 194   int (*idle_hook)(sasl_conn_t *conn);
 195   const sasl_callback_t *callbacks;
 196   const sasl_global_callbacks_t *global_callbacks; /* global callbacks
 197                                                     * connection */
 198   char *serverFQDN;

 200   /* Pointers to memory that we are responsible for */
 201   buffer_info_t *encode_buf;

 203   int error_code;
 204   char *error_buf, *errdetail_buf;
 205   size_t error_buf_len, errdetail_buf_len;
 206   char *mechlist_buf;
 207   size_t mechlist_buf_len;

 209   char *decode_buf;

 211   char user_buf[CANON_BUF_SIZE+1], authid_buf[CANON_BUF_SIZE+1];

 213 #ifdef _SUN_SDK_
 214   struct _sasl_global_context_s *gctx;
 215   /* EXPORT DELETE START */
 216   /* CRYPT DELETE START */
 215 #ifdef _INTEGRATED_SOLARIS_
 216   int sun_reg;
 217 #endif /* _INTEGRATED_SOLARIS_ */
 220   /* CRYPT DELETE END */
 221   /* EXPORT DELETE END */
 218 #endif /* _SUN_SDK_ */
 219 };
______unchanged_portion_omitted_
```



```
 228 #endif /* _SUN_SDK_ */

 230 /* Server Conn Type Information */

 232 typedef struct mechanism
 233 {
 234     int version;
 235     int condition; /* set to SASL_NOUSER if no available users;
 236                       set to SASL_CONTINUE if delayed plugn loading */
 237     char *plugname; /* for AUTHSOURCE tracking */
 238 #ifdef _SUN_SDK_
 243     /* EXPORT DELETE START */
 244     /* CRYPT DELETE START */
 239 #ifdef _INTEGRATED_SOLARIS_
 240     int sun_reg;
 241 #endif /* _INTEGRATED_SOLARIS_ */
 248     /* CRYPT DELETE END */
 249     /* EXPORT DELETE END */
 242     sasl_server_plug_t *plug;
 243         /*
 244          * The global context needs to be stored with separately from the
 245          * the plugin because it will be overwritten when the plugin is
 246          * reloaded
 247          */
 248     void *glob_context;
 249     struct mechanism *next;
 250 #else
 251     const sasl_server_plug_t *plug;
 252     struct mechanism *next;
 253     char *f;       /* where should i load the mechanism from? */
 254 #endif /* _SUN_SDK_ */
 255 } mechanism_t;
______unchanged_portion_omitted_

 285 /* Client Conn Type Information */

 287 typedef struct cmechanism
 288 {
 289     int version;

 291     char *plugname;
 292 #ifdef _SUN_SDK_
 301     /* EXPORT DELETE START */
 302     /* CRYPT DELETE START */
 293 #ifdef _INTEGRATED_SOLARIS_
 294     int sun_reg;
 295 #endif /* _INTEGRATED_SOLARIS_ */
 306     /* CRYPT DELETE END */
 307     /* EXPORT DELETE END */
 296         /*
 297          * The global context needs to be stored with separately from the
 298          * the plugin because it will be overwritten when the plugin is
 299          * reloaded
 300          */
 301     void *glob_context;
 302     sasl_client_plug_t *plug;
 303 #else
 304     const sasl_client_plug_t *plug;
 305 #endif /* _SUN_SDK_ */

 307     struct cmechanism *next;
 308 } cmechanism_t;
______unchanged_portion_omitted_

 421 /*
 422  * globals & constants
```

```
 */
/*
 * common.c
 */
#ifndef _SUN_SDK_
LIBSASL_API const sasl_utils_t *sasl_global_utils;

extern int (*_sasl_client_idle_hook)(sasl_conn_t *conn);
extern int (*_sasl_server_idle_hook)(sasl_conn_t *conn);

/* These return SASL_OK if we've actually finished cleanup,
 * SASL_NOTINIT if that part of the library isn't inited, and
 * SASL_CONTINUE if we need to call them again */
extern int (*_sasl_client_cleanup_hook)(void);
extern int (*_sasl_server_cleanup_hook)(void);

extern sasl_allocation_utils_t _sasl_allocation_utils;
extern sasl_mutex_utils_t _sasl_mutex_utils;
#endif /* !_SUN_SDK_ */

/*
 * checkpw.c
 */
extern struct sasl_verify_password_s _sasl_verify_password[];

/*
 * server.c
 */
/* (this is a function call to ensure this is read-only to the outside) */
#ifdef _SUN_SDK_
extern int _is_sasl_server_active(_sasl_global_context_t *gctx);
#else
extern int _is_sasl_server_active(void);
#endif /* _SUN_SDK_ */

/*
 * Allocation and Mutex utility macros
 */
#ifdef _SUN_SDK_
#define sasl_ALLOC(__size__) (gctx->sasl_allocation_utils.malloc((__size__)))
#define sasl_CALLOC(__nelem__, __size__) \
        (gctx->sasl_allocation_utils.calloc((__nelem__), (__size__)))
#define sasl_REALLOC(__ptr__, __size__) \
        (gctx->sasl_allocation_utils.realloc((__ptr__), (__size__)))
#define sasl_FREE(__ptr__) (gctx->sasl_allocation_utils.free((__ptr__)))
#define sasl_sun_ALLOC(__size__) (malloc((__size__)))
#define sasl_sun_CALLOC(__nelem__, __size__) (calloc((__nelem__), (__size__)))
#define sasl_sun_REALLOC(__ptr__, __size__) (realloc((__ptr__), (__size__)))
#define sasl_sun_FREE(__ptr__) (free((__ptr__)))

#define sasl_MUTEX_ALLOC() (gctx->sasl_mutex_utils.alloc())
#define sasl_MUTEX_LOCK(__mutex__) (gctx->sasl_mutex_utils.lock((__mutex__)))
#define sasl_MUTEX_UNLOCK(__mutex__) \
        (gctx->sasl_mutex_utils.unlock((__mutex__)))
#define sasl_MUTEX_FREE(__mutex__) (gctx->sasl_mutex_utils.free((__mutex__)))
#else
#define sasl_ALLOC(__size__) (_sasl_allocation_utils.malloc((__size__)))
#define sasl_CALLOC(__nelem__, __size__) \
        (_sasl_allocation_utils.calloc((__nelem__), (__size__)))
#define sasl_REALLOC(__ptr__, __size__) \
        (_sasl_allocation_utils.realloc((__ptr__), (__size__)))
#define sasl_FREE(__ptr__) (_sasl_allocation_utils.free((__ptr__)))

#define sasl_MUTEX_ALLOC() (_sasl_mutex_utils.alloc())
#define sasl_MUTEX_LOCK(__mutex__) (_sasl_mutex_utils.lock((__mutex__)))
#define sasl_MUTEX_UNLOCK(__mutex__) (_sasl_mutex_utils.unlock((__mutex__)))
```



```
#define sasl_MUTEX_FREE(__mutex__) \
        (_sasl_mutex_utils.free((__mutex__)))
#endif /* _SUN_SDK_ */

/* function prototypes */
/*
 * dlopen.c and staticopen.c
 */
/*
 * The differences here are:
 * _sasl_load_plugins loads all plugins from all files
 * _sasl_get_plugin loads the LIBRARY for an individual file
 * _sasl_done_with_plugins frees the LIBRARIES loaded by the above 2
 * _sasl_locate_entry locates an entrypoint in a given library
 */
#ifdef _SUN_SDK_
extern int _sasl_load_plugins(_sasl_global_context_t *gctx,
                              int server,
                              const add_plugin_list_t *entrypoints,
                              const sasl_callback_t *getpath_callback,
                              const sasl_callback_t *verifyfile_callback);

extern int _sasl_get_plugin(_sasl_global_context_t *gctx,
                            const char *file,
                            const sasl_callback_t *verifyfile_cb,
                            void **libraryptr);
extern int _sasl_locate_entry(void *library, const char *entryname,
                              void **entry_point);
extern int _sasl_done_with_plugins(_sasl_global_context_t *gctx);
#else
extern int _sasl_load_plugins(const add_plugin_list_t *entrypoints,
                              const sasl_callback_t *getpath_callback,
                              const sasl_callback_t *verifyfile_callback);
extern int _sasl_get_plugin(const char *file,
                            const sasl_callback_t *verifyfile_cb,
                            void **libraryptr);
extern int _sasl_locate_entry(void *library, const char *entryname,
                              void **entry_point);
extern int _sasl_done_with_plugins();
#endif /* _SUN_SDK_ */


/*
 * common.c
 */
extern const sasl_callback_t *
_sasl_find_getpath_callback(const sasl_callback_t *callbacks);

extern const sasl_callback_t *
_sasl_find_verifyfile_callback(const sasl_callback_t *callbacks);

#ifdef _SUN_SDK_
extern const sasl_callback_t *
_sasl_find_getconf_callback(const sasl_callback_t *callbacks);

extern int _sasl_common_init(_sasl_global_context_t *gctx,
                             sasl_global_callbacks_t *global_callbacks,
                             int server);
#else
extern int _sasl_common_init(sasl_global_callbacks_t *global_callbacks);
#endif /* _SUN_SDK_ */

extern int _sasl_conn_init(sasl_conn_t *conn,
                           const char *service,
                           unsigned int flags,
                           enum Sasl_conn_type type,
```

```
555                             int (*idle_hook)(sasl_conn_t *conn),
556                             const char *serverFQDN,
557                             const char *iplocalport,
558                             const char *ipremoteport,
559                             const sasl_callback_t *callbacks,
560                             const sasl_global_callbacks_t *global_callbacks);
561 extern void _sasl_conn_dispose(sasl_conn_t *conn);

563 #ifdef _SUN_SDK_
564 extern sasl_utils_t *
565 _sasl_alloc_utils(_sasl_global_context_t *gctx, sasl_conn_t *conn,
566                   sasl_global_callbacks_t *global_callbacks);
567 #else
568 extern sasl_utils_t *
569 _sasl_alloc_utils(sasl_conn_t *conn,
570                   sasl_global_callbacks_t *global_callbacks);
571 #endif /* _SUN_SDK_ */
572 extern int _sasl_free_utils(const sasl_utils_t ** utils);

574 extern int
575 _sasl_getcallback(sasl_conn_t * conn,
576                   unsigned long callbackid,
577                   int (**pproc)(),
578                   void **pcontext);

580 extern void
581 _sasl_log(sasl_conn_t *conn,
582           int level,
583           const char *fmt,
584           ...);

586 #ifdef _SUN_SDK_
587 extern void
588 __sasl_log(const _sasl_global_context_t *gctx,
589            const sasl_callback_t *callbacks,
590            int level,
591            const char *fmt,
592            ...);
593 #endif /* _SUN_SDK_ */
594 void _sasl_get_errorbuf(sasl_conn_t *conn, char ***bufhdl, size_t **lenhdl);
595 #ifdef _SUN_SDK_
596 int __sasl_add_string(const _sasl_global_context_t *gctx, char **out,
597                       size_t *alloclen,
598                       size_t *outlen, const char *add);

600 #define _sasl_add_string(out, alloclen, outlen, add) \
601         __sasl_add_string(gctx, out, alloclen, outlen, add)

603 /* More Generic Utilities in common.c */
604 #define _sasl_strdup(in, out, outlen) \
605         __sasl_strdup(gctx, in, out, outlen)
606 extern int __sasl_strdup(const _sasl_global_context_t *gctx, const char *in,
607                          char **out, size_t *outlen);

609 /* Basically a conditional call to realloc(), if we need more */
610 int __buf_alloc(const _sasl_global_context_t *gctx, char **rwbuf,
611         size_t *curlen, size_t newlen);
612 #define _buf_alloc(rwbuf, curlen, newlen) \
613         __buf_alloc(gctx, rwbuf, curlen, newlen)
614 #else
615 int _sasl_add_string(char **out, size_t *alloclen,
616                      size_t *outlen, const char *add);

618 /* More Generic Utilities in common.c */
619 extern int _sasl_strdup(const char *in, char **out, size_t *outlen);
```



```
621 /* Basically a conditional call to realloc(), if we need more */
622 int _buf_alloc(char **rwbuf, size_t *curlen, size_t newlen);
623 #endif /* _SUN_SDK_ */

625 /* convert an iovec to a single buffer */
626 #ifdef _SUN_SDK_
627 int _iovec_to_buf(const _sasl_global_context_t *gctx, const struct iovec *vec,
628                   unsigned numiov, buffer_info_t **output);
629 #else
630 int _iovec_to_buf(const struct iovec *vec,
631                   unsigned numiov, buffer_info_t **output);
632 #endif /* _SUN_SDK_ */

634 /* Convert between string formats and sockaddr formats */
635 int _sasl_iptostring(const struct sockaddr *addr, socklen_t addrlen,
636                      char *out, unsigned outlen);
637 int _sasl_ipfromstring(const char *addr, struct sockaddr *out,
638                        socklen_t outlen);

640 /*
641  * external plugin (external.c)
642  */
643 int external_client_plug_init(const sasl_utils_t *utils,
644                               int max_version,
645                               int *out_version,
646                               sasl_client_plug_t **pluglist,
647                               int *plugcount);
648 int external_server_plug_init(const sasl_utils_t *utils,
649                               int max_version,
650                               int *out_version,
651                               sasl_server_plug_t **pluglist,
652                               int *plugcount);

654 /* Mech Listing Functions */
655 #ifdef _SUN_SDK_
656 int _sasl_build_mechlist(_sasl_global_context_t *gctx);
657 #else
658 int _sasl_build_mechlist(void);
659 #endif /* _SUN_SDK_ */

661 int _sasl_server_listmech(sasl_conn_t *conn,
662                           const char *user,
663                           const char *prefix,
664                           const char *sep,
665                           const char *suffix,
666                           const char **result,
667                           unsigned *plen,
668                           int *pcount);
669 int _sasl_client_listmech(sasl_conn_t *conn,
670                           const char *prefix,
671                           const char *sep,
672                           const char *suffix,
673                           const char **result,
674                           unsigned *plen,
675                           int *pcount);
676 /* Just create a straight list of them */
677 #ifdef _SUN_SDK_
678 sasl_string_list_t *_sasl_client_mechs(_sasl_global_context_t *gctx);
679 sasl_string_list_t *_sasl_server_mechs(_sasl_global_context_t *gctx);
680 #else
681 sasl_string_list_t *_sasl_client_mechs(void);
682 sasl_string_list_t *_sasl_server_mechs(void);
683 #endif /* _SUN_SDK_ */

685 /*
686  * config file declarations (config.c)
```

```
 */
#ifdef _SUN_SDK_
extern int sasl_config_init(_sasl_global_context_t *gctx,
        const char *filename);
extern void sasl_config_free(_sasl_global_context_t *gctx);
extern const char *sasl_config_getstring(_sasl_global_context_t *gctx,
        const char *key,const char *def);
extern int sasl_config_getint(_sasl_global_context_t *gctx,
        const char *key,int def);
extern int sasl_config_getswitch(_sasl_global_context_t *gctx,
        const char *key,int def);
#else
extern int sasl_config_init(const char *filename);
extern const char *sasl_config_getstring(const char *key,const char *def);
extern int sasl_config_getint(const char *key,int def);
extern int sasl_config_getswitch(const char *key,int def);
#endif /* _SUN_SDK_ */

/* checkpw.c */
#ifdef DO_SASL_CHECKAPOP
extern int _sasl_auxprop_verify_apop(sasl_conn_t *conn,
                                     const char *userstr,
                                     const char *challenge,
                                     const char *response,
                                     const char *user_realm);
#endif /* DO_SASL_CHECKAPOP */

/* Auxprop Plugin (checkpw.c) */
extern int sasldb_auxprop_plug_init(const sasl_utils_t *utils,
                                    int max_version,
                                    int *out_version,
                                    sasl_auxprop_plug_t **plug,
                                    const char *plugname);

/*
 * auxprop.c
 */
#ifdef _SUN_SDK_
extern void _sasl_auxprop_free(_sasl_global_context_t *gctx);
#else
extern int _sasl_auxprop_add_plugin(void *p, void *library);
extern void _sasl_auxprop_free(void);
#endif /* _SUN_SDK_ */
extern void _sasl_auxprop_lookup(sasl_server_params_t *sparams,
                                 unsigned flags,
                                 const char *user, unsigned ulen);

/*
 * canonusr.c
 */
#ifdef _SUN_SDK_
void _sasl_canonuser_free(_sasl_global_context_t *gctx);
#else
void _sasl_canonuser_free();
#endif /* _SUN_SDK_ */
extern int internal_canonuser_init(const sasl_utils_t *utils,
                                   int max_version,
                                   int *out_version,
                                   sasl_canonuser_plug_t **plug,
                                   const char *plugname);
extern int _sasl_canon_user(sasl_conn_t *conn,
                            const char *user, unsigned ulen,
                            unsigned flags,
                            sasl_out_params_t *oparams);

#ifdef _SUN_SDK_
```



```
/* Private functions to create, free, and use a private context */
void *sasl_create_context(void);

void sasl_free_context(void *context);

extern int _sasl_server_init(void *ctx, const sasl_callback_t *callbacks,
                             const char *appname);

extern int _sasl_server_new(void *ctx, const char *service,
                            const char *serverFQDN, const char *user_realm,
                            const char *iplocalport, const char *ipremoteport,
                            const sasl_callback_t *callbacks, unsigned flags,
                            sasl_conn_t **pconn);

extern int _sasl_client_init(void *ctx,
                             const sasl_callback_t *callbacks);

extern int _sasl_client_new(void *ctx,
                            const char *service,
                            const char *serverFQDN,
                            const char *iplocalport,
                            const char *ipremoteport,
                            const sasl_callback_t *prompt_supp,
                            unsigned flags,
                            sasl_conn_t **pconn);

extern int _sasl_client_add_plugin(void *ctx,
                                   const char *plugname,
                                   sasl_client_plug_init_t *cplugfunc);
extern int _sasl_server_add_plugin(void *ctx,
                                   const char *plugname,
                                   sasl_server_plug_init_t *splugfunc);
extern int _sasl_canonuser_add_plugin(void *ctx,
                                      const char *plugname,
                                      sasl_canonuser_init_t *canonuserfunc);
extern int _sasl_auxprop_add_plugin(void *ctx,
                                    const char *plugname,
                                    sasl_auxprop_init_t *auxpropfunc);

_sasl_global_context_t *_sasl_gbl_ctx(void);

/* EXPORT DELETE START */
/* CRYPT DELETE START */
#ifdef _INTEGRATED_SOLARIS_
int _is_sun_reg(void *mech);
#endif /* _INTEGRATED_SOLARIS_ */
/* CRYPT DELETE END */
/* EXPORT DELETE END */

/* unsupported functions that are used internally */
int sasl_randcreate(sasl_rand_t **rpool);

void sasl_randfree(sasl_rand_t **rpool);

void sasl_rand(sasl_rand_t *rpool, char *buf, unsigned len);

void sasl_churn(sasl_rand_t *rpool, const char *data, unsigned len);

int sasl_mkchal(sasl_conn_t *conn, char *buf, unsigned maxlen,
                unsigned hostflag);
#endif  /* _SUN_SDK_ */

#endif /* SASLINT_H */
```

```
**********************************************************
   69720 Thu Jul 11 01:29:32 2013
new/usr/src/lib/libsasl/lib/server.c
first pass
**********************************************************
______unchanged_portion_omitted_

 379 int _sasl_server_add_plugin(void *ctx,
 380                             const char *plugname,
 381                             sasl_server_plug_init_t *p)
 382 {
 383     int nplug = 0;
 384     int i;
 385     mechanism_t *m;
 386     _sasl_global_context_t *gctx = ctx == NULL ? _sasl_gbl_ctx() : ctx;
 387     mech_list_t *mechlist = gctx->mechlist;

 389     /* EXPORT DELETE START */
 390     /* CRYPT DELETE START */
 389 #ifdef _INTEGRATED_SOLARIS_
 390     int sun_reg;
 391 #endif /* _INTEGRATED_SOLARIS_ */
 394     /* CRYPT DELETE END */
 395     /* EXPORT DELETE END */
 392 #else
 393 {
 394 #endif /* _SUN_SDK_ */
 395     int plugcount;
 396     sasl_server_plug_t *pluglist;
 397     mechanism_t *mech;
 398     sasl_server_plug_init_t *entry_point;
 399     int result;
 400     int version;
 401     int lupe;

 403     if(!plugname || !p) return SASL_BADPARAM;

 405 #ifdef _SUN_SDK_
 406     if (mechlist == NULL) return SASL_BADPARAM;

 408     /* Check to see if this plugin has already been registered */
 409     m = mechlist->mech_list;
 410     for (i = 0; i < mechlist->mech_length; i++) {
 411         if (strcmp(plugname, m->plugname) == 0)
 412             return SASL_OK;
 413         m = m->next;
 414     }

 416     result = LOCK_MUTEX(&server_plug_mutex);
 417     if (result != SASL_OK)
 418         return result;

 420 #endif /* _SUN_SDK_ */
 421     entry_point = (sasl_server_plug_init_t *)p;

 423     /* call into the shared library asking for information about it */
 424     /* version is filled in with the version of the plugin */
 425     result = entry_point(mechlist->utils, SASL_SERVER_PLUG_VERSION, &version,
 426                          &pluglist, &plugcount);

 432     /* EXPORT DELETE START */
 433     /* CRYPT DELETE START */
 428 #ifdef _INTEGRATED_SOLARIS_
 429     sun_reg = _is_sun_reg(pluglist);
 430 #endif /* _INTEGRATED_SOLARIS_ */
 437     /* CRYPT DELETE END */
```



```
 438     /* EXPORT DELETE END */

 432 #ifdef _SUN_SDK_
 433     if (result != SASL_OK) {
 434         UNLOCK_MUTEX(&server_plug_mutex);
 435         __sasl_log(gctx, gctx->server_global_callbacks.callbacks,
 436                    SASL_LOG_DEBUG,
 437                    "server add_plugin entry_point error %z", result);
 438 #else
 439     if ((result != SASL_OK) && (result != SASL_NOUSER)) {
 440         _sasl_log(NULL, SASL_LOG_DEBUG,
 441                   "server add_plugin entry_point error %z\n", result);
 442 #endif /* _SUN_SDK_ */
 443         return result;
 444     }

 446     /* Make sure plugin is using the same SASL version as us */
 447     if (version != SASL_SERVER_PLUG_VERSION)
 448     {
 449 #ifdef _SUN_SDK_
 450         UNLOCK_MUTEX(&server_plug_mutex);
 451         __sasl_log(gctx, gctx->server_global_callbacks.callbacks,
 452                    SASL_LOG_ERR, "version mismatch on plugin");
 453 #else
 454         _sasl_log(NULL, SASL_LOG_ERR,
 455                   "version mismatch on plugin");
 456 #endif /* _SUN_SDK_ */
 457         return SASL_BADVERS;
 458     }
 459 #ifdef _SUN_SDK_
 460     /* Check plugins to make sure mech_name is non-NULL */
 461     for (lupe=0;lupe < plugcount ;lupe++) {
 462         if (pluglist[lupe].mech_name == NULL)
 463             break;
 464     }
 465     if (lupe < plugcount) {
 466 #ifdef _SUN_SDK_
 467         UNLOCK_MUTEX(&server_plug_mutex);
 468         __sasl_log(gctx, gctx->server_global_callbacks.callbacks,
 469                    SASL_LOG_ERR, "invalid server plugin %s", plugname);
 470 #else
 471         _sasl_log(NULL, SASL_LOG_ERR, "invalid server plugin %s", plugname);
 472 #endif /* _SUN_SDK_ */
 473         return SASL_BADPROT;
 474     }
 475 #endif /* _SUN_SDK_ */

 477     for (lupe=0;lupe < plugcount ;lupe++)
 478     {
 479 #ifdef _SUN_SDK_
 480         if (!load_mech(gctx, pluglist->mech_name)) {
 481             pluglist++;
 482             continue;
 483         }
 484         nplug++;
 485 #endif /* _SUN_SDK_ */
 486         mech = sasl_ALLOC(sizeof(mechanism_t));
 487 #ifdef _SUN_SDK_
 488         if (! mech) {
 489             UNLOCK_MUTEX(&server_plug_mutex);
 490             return SASL_NOMEM;
 491         }

 493         mech->glob_context = pluglist->glob_context;
 494 #else
 495         if (! mech) return SASL_NOMEM;
```

```
 496 #endif /* _SUN_SDK_ */

 498         mech->plug=pluglist++;
 499         if(_sasl_strdup(plugname, &mech->plugname, NULL) != SASL_OK) {
 500 #ifdef _SUN_SDK_
 501             UNLOCK_MUTEX(&server_plug_mutex);
 502 #endif /* _SUN_SDK_ */
 503             sasl_FREE(mech);
 504             return SASL_NOMEM;
 505         }
 506         mech->version = version;
 507 #ifdef _SUN_SDK_
 516         /* EXPORT DELETE START */
 517         /* CRYPT DELETE START */
 508 #ifdef _INTEGRATED_SOLARIS_
 509         mech->sun_reg = sun_reg;
 510 #endif /* _INTEGRATED_SOLARIS_ */
 521         /* CRYPT DELETE END */
 522         /* EXPORT DELETE END */

 512         /* whether this mech actually has any users in it's db */
 513         mech->condition = SASL_OK;
 514 #else
 515         /* whether this mech actually has any users in it's db */
 516         mech->condition = result; /* SASL_OK or SASL_NOUSER */
 517 #endif /* _SUN_SDK_ */

 519         mech->next = mechlist->mech_list;
 520         mechlist->mech_list = mech;
 521         mechlist->mech_length++;
 522     }

 524 #ifdef _SUN_SDK_
 525     UNLOCK_MUTEX(&server_plug_mutex);
 526     return (nplug == 0) ? SASL_NOMECH : SASL_OK;
 527 #else
 528     return SASL_OK;
 529 #endif /* _SUN_SDK_ */
 530 }
_____unchanged_portion_omitted_

1403 /*
1404  * The rule is:
1405  * IF mech strength + external strength < min ssf THEN FAIL
1406  * We also have to look at the security properties and make sure
1407  * that this mechanism has everything we want
1408  */
1409 static int mech_permitted(sasl_conn_t *conn,
1410                           mechanism_t *mech)
1411 {
1412     sasl_server_conn_t *s_conn = (sasl_server_conn_t *)conn;
1413     const sasl_server_plug_t *plug;
1414     int myflags;
1415     context_list_t *cur;
1416     sasl_getopt_t *getopt;
1417     void *context;
1418     sasl_ssf_t minssf = 0;
1419 #ifdef _SUN_SDK_
1420     _sasl_global_context_t *gctx;
1421 #endif /* _SUN_SDK_ */

1423     if(!conn) return 0;

1425 #ifdef _SUN_SDK_
1426     gctx = conn->gctx;
1427 #endif /* _SUN_SDK_ */
```



```
1429     if(! mech || ! mech->plug) {
1430 #ifdef _SUN_SDK_
1431         if(conn) _sasl_log(conn, SASL_LOG_WARN, "Parameter error");
1432 #else
1433         PARAMERROR(conn);
1434 #endif /* _SUN_SDK_ */
1435         return 0;
1436     }
1437
1438     plug = mech->plug;

1440     /* get the list of allowed mechanisms (default = all) */
1441     if (_sasl_getcallback(conn, SASL_CB_GETOPT, &getopt, &context)
1442             == SASL_OK) {
1443         const char *mlist = NULL;

1445         getopt(context, NULL, "mech_list", &mlist, NULL);

1447         /* if we have a list, check the plugin against it */
1448         if (mlist) {
1449             const char *cp;

1451             while (*mlist) {
1452                 for (cp = mlist; *cp && !isspace((int) *cp); cp++);
1453                 if (((size_t) (cp - mlist) == strlen(plug->mech_name)) &&
1454                     !strncasecmp(mlist, plug->mech_name,
1455                                  strlen(plug->mech_name))) {
1456                     break;
1457                 }
1458                 mlist = cp;
1459                 while (*mlist && isspace((int) *mlist)) mlist++;
1460             }

1462             if (!*mlist) return 0;  /* reached EOS -> not in our list */
1463         }
1464     }

1466     /* setup parameters for the call to mech_avail */
1467     s_conn->sparams->serverFQDN=conn->serverFQDN;
1468     s_conn->sparams->service=conn->service;
1469     s_conn->sparams->user_realm=s_conn->user_realm;
1470     s_conn->sparams->props=conn->props;
1471     s_conn->sparams->external_ssf=conn->external.ssf;

1473     /* Check if we have banished this one already */
1474     for(cur = s_conn->mech_contexts; cur; cur=cur->next) {
1475         if(cur->mech == mech) {
1476             /* If it's not mech_avail'd, then stop now */
1477             if(!cur->context) return 0;
1478             break;
1479         }
1480     }
1481
1494     /* EXPORT DELETE START */
1495     /* CRYPT DELETE START */
1482 #ifdef _INTEGRATED_SOLARIS_
1483     if (!mech->sun_reg) {
1484         s_conn->sparams->props.min_ssf = 0;
1485         s_conn->sparams->props.max_ssf = 0;
1486     }
1487     s_conn->base.sun_reg = mech->sun_reg;
1488 #endif /* _INTEGRATED_SOLARIS_ */
1503     /* CRYPT DELETE END */
1504     /* EXPORT DELETE END */
1489     if (conn->props.min_ssf < conn->external.ssf) {
```

```
1490         minssf = 0;
1491     } else {
1492         minssf = conn->props.min_ssf - conn->external.ssf;
1493     }
1494
1495     /* Generic mechanism */
1512     /* EXPORT DELETE START */
1513     /* CRYPT DELETE START */
1496 #ifdef _INTEGRATED_SOLARIS_
1497     /* If not SUN supplied mech, it has no strength */
1498     if (plug->max_ssf < minssf || (minssf > 0 && !mech->sun_reg)) {
1499 #else
1518     /* CRYPT DELETE END */
1519     /* EXPORT DELETE END */
1500     if (plug->max_ssf < minssf) {
1521     /* EXPORT DELETE START */
1522     /* CRYPT DELETE START */
1501 #endif /* _INTEGRATED_SOLARIS_ */
1524     /* CRYPT DELETE END */
1525     /* EXPORT DELETE END */
1502 #ifdef _INTEGRATED_SOLARIS_
1503         sasl_seterror(conn, SASL_NOLOG,
1504                       gettext("mech %s is too weak"), plug->mech_name);
1505 #else
1506         sasl_seterror(conn, SASL_NOLOG,
1507                       "mech %s is too weak", plug->mech_name);
1508 #endif /* _INTEGRATED_SOLARIS_ */
1509         return 0; /* too weak */
1510     }

1512     context = NULL;
1513     if(plug->mech_avail
1514 #ifdef _SUN_SDK_
1515        && plug->mech_avail(mech->glob_context,
1516 #else
1517        && plug->mech_avail(plug->glob_context,
1518 #endif /* _SUN_SDK_ */
1519                            s_conn->sparams, (void **)&context) != SASL_OK ) {
1520         /* Mark this mech as no good for this connection */
1521         cur = sasl_ALLOC(sizeof(context_list_t));
1522         if(!cur) {
1523 #ifdef _SUN_SDK_
1524             if(conn) _sasl_log(conn, SASL_LOG_WARN, "Out of Memory");
1525 #else
1526             MEMERROR(conn);
1527 #endif /* _SUN_SDK_ */
1528             return 0;
1529         }
1530         cur->context = NULL;
1531         cur->mech = mech;
1532         cur->next = s_conn->mech_contexts;
1533         s_conn->mech_contexts = cur;

1535         /* Error should be set by mech_avail call */
1536         return 0;
1537     } else if(context) {
1538         /* Save this context */
1539         cur = sasl_ALLOC(sizeof(context_list_t));
1540         if(!cur) {
1541 #ifdef _SUN_SDK_
1542             if(conn) _sasl_log(conn, SASL_LOG_WARN, "Out of Memory");
1543 #else
1544             MEMERROR(conn);
1545 #endif /* _SUN_SDK_ */
1546             return 0;
1547         }
```



```
1548         cur->context = context;
1549         cur->mech = mech;
1550         cur->next = s_conn->mech_contexts;
1551         s_conn->mech_contexts = cur;
1552     }
1553
1554     /* Generic mechanism */
1579     /* EXPORT DELETE START */
1580     /* CRYPT DELETE START */
1555 #ifdef _INTEGRATED_SOLARIS_
1556     /* If not SUN supplied mech, it has no strength */
1557     if (plug->max_ssf < minssf || (minssf > 0 && !mech->sun_reg)) {
1558 #else
1585     /* CRYPT DELETE END */
1586     /* EXPORT DELETE END */
1559     if (plug->max_ssf < minssf) {
1588     /* EXPORT DELETE START */
1589     /* CRYPT DELETE START */
1560 #endif /* _INTEGRATED_SOLARIS_ */
1591     /* CRYPT DELETE END */
1592     /* EXPORT DELETE END */
1561 #ifdef _INTEGRATED_SOLARIS_
1562         sasl_seterror(conn, SASL_NOLOG, gettext("too weak"));
1563 #else
1564         sasl_seterror(conn, SASL_NOLOG, "too weak");
1565 #endif /* _INTEGRATED_SOLARIS_ */
1566         return 0; /* too weak */
1567     }

1569 #ifndef _SUN_SDK_
1570     /* if there are no users in the secrets database we can't use this
1571        mechanism */
1572     if (mech->condition == SASL_NOUSER) {
1573         sasl_seterror(conn, 0, "no users in secrets db");
1574         return 0;
1575     }
1576 #endif /* !_SUN_SDK_ */

1578     /* Can it meet our features? */
1579     if ((conn->flags & SASL_NEED_PROXY) &&
1580         !(plug->features & SASL_FEAT_ALLOWS_PROXY)) {
1581         return 0;
1582     }
1583
1584     /* security properties---if there are any flags that differ and are
1585        in what the connection are requesting, then fail */
1586
1587     /* special case plaintext */
1588     myflags = conn->props.security_flags;

1590     /* if there's an external layer this is no longer plaintext */
1591     if ((conn->props.min_ssf <= conn->external.ssf) &&
1592         (conn->external.ssf > 1)) {
1593         myflags &= ~SASL_SEC_NOPLAINTEXT;
1594     }

1596     /* do we want to special case SASL_SEC_PASS_CREDENTIALS? nah.. */
1597     if (((myflags ^ plug->security_flags) & myflags) != 0) {
1598 #ifdef _INTEGRATED_SOLARIS_
1599         sasl_seterror(conn, SASL_NOLOG,
1600                       gettext("security flags do not match required"));
1601 #else
1602         sasl_seterror(conn, SASL_NOLOG,
1603                       "security flags do not match required");
1604 #endif /* _INTEGRATED_SOLARIS_ */
1605         return 0;
```

```
1606     }

1608     /* Check Features */
1609     if(plug->features & SASL_FEAT_GETSECRET) {
1610         /* We no longer support sasl_server_{get,put}secret */
1611 #ifdef _SUN_SDK_
1612         _sasl_log(conn, SASL_LOG_ERR,
1613                   "mech %s requires unprovided secret facility",
1614                   plug->mech_name);
1615 #else
1616         sasl_seterror(conn, 0,
1617                       "mech %s requires unprovided secret facility",
1618                       plug->mech_name);
1619 #endif /* _SUN_SDK_ */
1620         return 0;
1621     }

1623     return 1;
1624 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1862 Thu Jul 11 01:29:33 2013
new/usr/src/lib/libsldap/Makefile
first pass
**********************************************************
```

```
#
# 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 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# ident "%Z%%M% %I%     %E% SMI"
#

include                 ../Makefile.lib

SUBDIRS =       $(MACH)
$(BUILD64)SUBDIRS += $(MACH64)

all :=          TARGET= all
clean :=        TARGET= clean
clobber :=      TARGET= clobber
delete :=       TARGET= delete
install :=      TARGET= install
lint :=         TARGET= lint
catalog :=      TARGET= catalog
_msg :=         TARGET= _msg
package :=      TARGET= package

LIBRARY=        libsldap.a
TEXT_DOMAIN =   SUNW_OST_OSLIB
XGETFLAGS=
POFILE=         $(LIBRARY:.a=.po)
POFILES=        generic.po

.KEEP_STATE:

all clean clobber delete install lint catalog package: $(SUBDIRS)

# install rule for install_h target
$(ROOTHDRDIR)/%: %
        $(INS.file)

$(SUBDIRS):     FRC
        @cd $@; pwd; $(MAKE) $(TARGET)

_msg:   $(MSGDOMAIN) $(POFILE)
        $(RM) $(MSGDOMAIN)/$(POFILE)
```



```
        $(RM) generic.po
        cp $(POFILE) $(MSGDOMAIN)

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

$(MSGDOMAIN):
        $(INS.dir)

$(POFILES):
        $(RM) messages.po
        $(XGETTEXT) $(XGETFLAGS)  common/*.[ch]*
        sed "/^domain/d" < messages.po  > $@
        $(RM) messages.po

FRC:

# EXPORT DELETE START
EXPORT_SRC:
        $(RM) -f common/ns_crypt.c+ Makefile+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" < \
                common/ns_crypt.c > common/ns_crypt.c+
        $(MV) common/ns_crypt.c+ common/ns_crypt.c
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile common/ns_crypt.c

# EXPORT DELETE END
```

```
**********************************************************
    4801 Thu Jul 11 01:29:34 2013
new/usr/src/lib/libsldap/common/ns_crypt.c
pass 2
first pass
**********************************************************
```

```
/*
 * 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 1999-2003 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */

#pragma ident   "%Z%%M% %I%     %E% SMI"

#include <stdlib.h>
#include <string.h>
#include <libintl.h>
#include <locale.h>
#include <errno.h>
#include <unistd.h>
#include <ctype.h>
#include <syslog.h>
#include <sys/time.h>
#include "ns_sldap.h"
#include "ns_internal.h"
/* EXPORT DELETE START */
#include <crypt.h>

#define NS_DOMESTIC     1

static  char            t1[ROTORSIZE];
static  char            t2[ROTORSIZE];
static  char            t3[ROTORSIZE];
static  char            hexdig[] = "0123456789abcdef";

static mutex_t          ns_crypt_lock = DEFAULTMUTEX;
static boolean_t        crypt_inited = B_FALSE;

static int
is_cleartext(const char *pwd)
{
        if (0 == strncmp(pwd, CRYPTMARK, strlen(CRYPTMARK)))
                return (FALSE);
```



```
        return (TRUE);
}
_____unchanged_portion_omitted_
/* EXPORT DELETE END */


static void
c_setup()
{
/* EXPORT DELETE START */
        int ic, i, k, temp;
        unsigned random;
        char buf[13];
        int seed;

        (void) mutex_lock(&ns_crypt_lock);
        if (crypt_inited) {
                (void) mutex_unlock(&ns_crypt_lock);
                return;
        }
        (void) strcpy(buf, "Homer J");
        buf[8] = buf[0];
        buf[9] = buf[1];
        (void) strncpy(buf, (char *)crypt(buf, &buf[8]), 13);
        seed = 123;
        for (i = 0; i < 13; i++)
                seed = seed*buf[i] + i;
        for (i = 0; i < ROTORSIZE; i++) {
                t1[i] = i;
                t3[i] = 0;
        }
        for (i = 0; i < ROTORSIZE; i++) {
                seed = 5*seed + buf[i%13];
                random = seed % 65521;
                k = ROTORSIZE-1 - i;
                ic = (random&MASK)%(k+1);
                random >>= 8;
                temp = t1[k];
                t1[k] = t1[ic];
                t1[ic] = temp;
                if (t3[k] != 0) continue;
                ic = (random&MASK) % k;
                while (t3[ic] != 0) ic = (ic + 1) % k;
                t3[k] = ic;
                t3[ic] = k;
        }
        for (i = 0; i < ROTORSIZE; i++)
                t2[t1[i]&MASK] = i;
        crypt_inited = B_TRUE;
        (void) mutex_unlock(&ns_crypt_lock);
}


static char *
modvalue(char *str, int len, int *mod_len)
{
        int i, n1, n2;
        char *s;

        if (!crypt_inited)
                c_setup();
        i = 0;
        n1 = 0;
        n2 = 0;
        if ((s = (char *)malloc(2 * len + 1)) != NULL) {
                while (i < len) {
```

```
                    s[i] = t2[(t3[(t1[(str[i]+n1)&MASK]+n2)&MASK]-n2)&MASK]-n1;
                    i++;
                    n1++;
                    if (n1 == ROTORSIZE) {
                        n1 = 0;
                        n2++;
                        if (n2 == ROTORSIZE) n2 = 0;
                    }
                }
                s[i] = '\0';
                if (mod_len != NULL)
                    *mod_len = i;
        }
        return (s);
/* EXPORT DELETE END */
}


char *
evalue(char *ptr)
{
/* EXPORT DELETE START */
        char *modv, *str, *ev;
        int modv_len;
        size_t len;

        /*
         * if not cleartext, return a copy of what ptr
         * points to as that is what evalue does below.
         */
        if (FALSE == is_cleartext(ptr)) {
                str = strdup(ptr);
                return (str);
        }

        modv = modvalue(ptr, strlen(ptr), &modv_len);
        str = hex2ascii(modv, modv_len);
        free(modv);
        modv = NULL;
        len = strlen(str) + strlen(CRYPTMARK) + 1;
        ev = malloc(len);
        if (ev == NULL) {
                free(str);
                return (NULL);
        }
        (void) snprintf(ev, len, CRYPTMARK "%s", str);
        free(str);
        str = NULL;
        return (ev);
#ifndef NS_DOMESTIC
/* EXPORT DELETE END */
        return (strdup(ptr));
/* EXPORT DELETE START */
#endif
/* EXPORT DELETE END */
}


char *
dvalue(char *ptr)
{
/* EXPORT DELETE START */
        char *modv, *str, *sb;
        int len;

        /* if cleartext return NULL (error!) */
```



```
        if (TRUE == is_cleartext(ptr))
                return (NULL);

        sb = strchr(ptr, '}');
        sb++;
        len = strlen(sb);
        str = ascii2hex(sb, &len);
        modv = modvalue(str, len, NULL);
        free(str);
        str = NULL;
        return (modv);
#ifndef NS_DOMESTIC
/* EXPORT DELETE END */
        return (strdup(ptr));
/* EXPORT DELETE START */
#endif
/* EXPORT DELETE END */
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    1511 Thu Jul 11 01:29:34 2013
new/usr/src/lib/pam_modules/krb5/Makefile
first pass
**********************************************************
```

```
#
# 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 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# ident "%Z%%M% %I%     %E% SMI"
#

include         ../../Makefile.lib

TEXT_DOMAIN=    SUNW_OST_SYSOSPAM
POFILE=         pam_krb5.po
MSGFILES=       krb5_acct_mgmt.c krb5_authenticate.c krb5_password.c \
                krb5_setcred.c

SUBDIRS=        $(MACH)
$(BUILD64)SUBDIRS +=    $(MACH64)

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

# EXPORT DELETE START
# CRYPT DELETE START
EXPORT_SRC :=   TARGET= EXPORT_SRC
CRYPT_SRC :=    TARGET= CRYPT_SRC
# CRYPT DELETE END
# EXPORT DELETE END

.KEEP_STATE:

all clean clobber install lint: $(SUBDIRS)

$(POFILE):      $(MSGFILES)
        $(BUILDPO.msgfiles)

_msg:           $(MSGDOMAINPOFILE)

$(SUBDIRS):     FRC
        @cd $@; pwd; $(MAKE) $(TARGET)
```



```
# EXPORT DELETE START
# CRYPT DELETE START
# Special target to clean up the source tree for export distribution
# Warning: This target changes the source tree
EXPORT_SRC:
        $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d"    \
                < Makefile.com > Makefile.com+
        $(MV) Makefile.com+ Makefile.com
        $(CHMOD) 444 Makefile.com
        $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d"    \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile

CRYPT_SRC:
        $(SED) -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d"      \
                < Makefile.com \
            | $(SED) -e "/EXPORT DELETE/d" \
                > Makefile.com+
        $(MV) Makefile.com+ Makefile.com
        $(CHMOD) 444 Makefile.com
        $(SED) -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d"      \
                < Makefile \
            | $(SED) -e "/EXPORT DELETE/d" \
                > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile
# CRYPT DELETE END
# EXPORT DELETE END
```

```
FRC:

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

```
**********************************************************
    1078 Thu Jul 11 01:29:35 2013
new/usr/src/lib/pkcs11/pkcs11_softtoken/common/Makefile
first pass
**********************************************************
```

```
   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 # Copyright 2003 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #
  25 # ident "%Z%%M% %I%     %E% SMI"
  26 #
  27 # lib/pkcs11/pkcs11_softtoken/common/Makefile
  28 #
  29 # include global definitions
  30 include $(SRC)/Makefile.master

  32 .KEEP_STATE:

  34 FRC:

  36 # EXPORT DELETE START
  37 EXPORT_SRC:
  38         $(RM) Makefile+ softRSA.c+
  39         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  40                 < softRSA.c > softRSA.c+
  41         $(MV) softRSA.c+ softRSA.c
  42         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
  43                 < Makefile > Makefile+
  44         $(RM) Makefile
  45         $(MV) Makefile+ Makefile
  46         $(CHMOD) 444 Makefile softRSA.c
  47 # EXPORT DELETE END
```

```
**********************************************************
   32171 Thu Jul 11 01:29:36 2013
new/usr/src/lib/pkcs11/pkcs11_softtoken/common/softRSA.c
first pass
**********************************************************
```

```
/*
 * 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) 2003, 2010, Oracle and/or its affiliates. All rights reserved.
 */

#include <pthread.h>
#include <stdlib.h>
#include <string.h>
#include <strings.h>
#include <sys/types.h>
#include <security/cryptoki.h>
#include <cryptoutil.h>
#include "softGlobal.h"
#include "softSession.h"
#include "softObject.h"
#include "softOps.h"
#include "softRSA.h"
#include "softMAC.h"
#include "softCrypt.h"

CK_RV
soft_rsa_encrypt(soft_object_t *key, CK_BYTE_PTR in, uint32_t in_len,
    CK_BYTE_PTR out, int realpublic)
{

        CK_RV rv = CKR_OK;

/* EXPORT DELETE START */

        uchar_t expo[MAX_KEY_ATTR_BUFLEN];
        uchar_t modulus[MAX_KEY_ATTR_BUFLEN];
        uint32_t expo_len = sizeof (expo);
        uint32_t modulus_len = sizeof (modulus);
        RSAbytekey k;

        if (realpublic) {
                rv = soft_get_public_value(key, CKA_PUBLIC_EXPONENT, expo,
                    &expo_len);
                if (rv != CKR_OK) {
                        goto clean1;
                }
```



```
        } else {
                rv = soft_get_private_value(key, CKA_PRIVATE_EXPONENT, expo,
                    &expo_len);
                if (rv != CKR_OK) {
                        goto clean1;
                }
        }

        rv = soft_get_public_value(key, CKA_MODULUS, modulus, &modulus_len);
        if (rv != CKR_OK) {
                goto clean1;
        }

        k.modulus = modulus;
        k.modulus_bits = CRYPTO_BYTES2BITS(modulus_len);
        k.pubexpo = expo;
        k.pubexpo_bytes = expo_len;
        k.rfunc = NULL;

        rv = rsa_encrypt(&k, in, in_len, out);

clean1:

/* EXPORT DELETE END */

        return (rv);
}


CK_RV
soft_rsa_decrypt(soft_object_t *key, CK_BYTE_PTR in, uint32_t in_len,
    CK_BYTE_PTR out)
{

        CK_RV rv = CKR_OK;

/* EXPORT DELETE START */

        uchar_t modulus[MAX_KEY_ATTR_BUFLEN];
        uchar_t prime1[MAX_KEY_ATTR_BUFLEN];
        uchar_t prime2[MAX_KEY_ATTR_BUFLEN];
        uchar_t expo1[MAX_KEY_ATTR_BUFLEN];
        uchar_t expo2[MAX_KEY_ATTR_BUFLEN];
        uchar_t coef[MAX_KEY_ATTR_BUFLEN];
        uint32_t modulus_len = sizeof (modulus);
        uint32_t prime1_len = sizeof (prime1);
        uint32_t prime2_len = sizeof (prime2);
        uint32_t expo1_len = sizeof (expo1);
        uint32_t expo2_len = sizeof (expo2);
        uint32_t coef_len = sizeof (coef);
        RSAbytekey k;

        rv = soft_get_private_value(key, CKA_MODULUS, modulus, &modulus_len);
        if (rv != CKR_OK) {
                goto clean1;
        }

        rv = soft_get_private_value(key, CKA_PRIME_1, prime1, &prime1_len);

        if ((prime1_len == 0) && (rv == CKR_OK)) {
                rv = soft_rsa_encrypt(key, in, in_len, out, 0);
                goto clean1;
        } else {
                if (rv != CKR_OK)
                        goto clean1;
        }
```

```
123         rv = soft_get_private_value(key, CKA_PRIME_2, prime2, &prime2_len);

125         if ((prime2_len == 0) && (rv == CKR_OK)) {
126                 rv = soft_rsa_encrypt(key, in, in_len, out, 0);
127                 goto clean1;
128         } else {
129                 if (rv != CKR_OK)
130                         goto clean1;
131         }

133         rv = soft_get_private_value(key, CKA_EXPONENT_1, expo1, &expo1_len);

135         if ((expo1_len == 0) && (rv == CKR_OK)) {
136                 rv = soft_rsa_encrypt(key, in, in_len, out, 0);
137                 goto clean1;
138         } else {
139                 if (rv != CKR_OK)
140                         goto clean1;
141         }

143         rv = soft_get_private_value(key, CKA_EXPONENT_2, expo2, &expo2_len);

145         if ((expo2_len == 0) && (rv == CKR_OK)) {
146                 rv = soft_rsa_encrypt(key, in, in_len, out, 0);
147                 goto clean1;
148         } else {
149                 if (rv != CKR_OK)
150                         goto clean1;
151         }

153         rv = soft_get_private_value(key, CKA_COEFFICIENT, coef, &coef_len);

155         if ((coef_len == 0) && (rv == CKR_OK)) {
156                 rv = soft_rsa_encrypt(key, in, in_len, out, 0);
157                 goto clean1;
158         } else {
159                 if (rv != CKR_OK)
160                         goto clean1;
161         }

163         k.modulus = modulus;
164         k.modulus_bits = CRYPTO_BYTES2BITS(modulus_len);
165         k.prime1 = prime1;
166         k.prime1_bytes = prime1_len;
167         k.prime2 = prime2;
168         k.prime2_bytes = prime2_len;
169         k.expo1 = expo1;
170         k.expo1_bytes = expo1_len;
171         k.expo2 = expo2;
172         k.expo2_bytes = expo2_len;
173         k.coeff = coef;
174         k.coeff_bytes = coef_len;
175         k.rfunc = NULL;

177         rv = rsa_decrypt(&k, in, in_len, out);

179 clean1:

187 /* EXPORT DELETE END */

181         return (rv);
182 }
```

_____unchanged_portion_omitted_

```
**********************************************************
    1472 Thu Jul 11 01:29:37 2013
new/usr/src/lib/sasl_plugins/Makefile
first pass
**********************************************************
```

```
#
# 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 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
#

# Note, to build SASL msg file go to $SRC/lib/libsasl and make _msg
# target there.  Messages in sasl_plugins will be picked up from there.

include ../Makefile.lib

SUBDIRS =       cram digestmd5 gssapi plain login

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

.KEEP_STATE:

all clean clobber install lint: $(SUBDIRS)

install_h check:

$(SUBDIRS): FRC
        @cd $@; pwd; $(MAKE) $(TARGET)

FRC:

# EXPORT DELETE START
# CRYPT DELETE START
# Special target to clean up the source tree for export distribution
# Warning: This target changes the source tree
EXPORT_SRC:
        $(RM)   Makefile+ \
                digestmd5/digestmd5.c+ \
                gssapi/gssapi.c+

        $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
```



```
        $(MV) Makefile+ Makefile

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < digestmd5/digestmd5.c > digestmd5/digestmd5.c+
        $(MV) digestmd5/digestmd5.c+ digestmd5/digestmd5.c

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < gssapi/gssapi.c > gssapi/gssapi.c+
        $(MV) gssapi/gssapi.c+ gssapi/gssapi.c

        $(CHMOD) 444 \
                Makefile \
                digestmd5/digestmd5.c \
                gssapi/gssapi.c

CRYPT_SRC:
        $(RM) Makefile+

        $(SED) -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
                < digestmd5/digestmd5.c | $(SED) -e "/EXPORT DELETE/d" \
                > digestmd5/digestmd5.c+
        $(MV) digestmd5/digestmd5.c+ digestmd5/digestmd5.c

        $(SED) -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
                < gssapi/gssapi.c | $(SED) -e "/EXPORT DELETE/d" \
                > gssapi/gssapi.c+
        $(MV) gssapi/gssapi.c+ gssapi/gssapi.c

        $(SED) -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d" \
                < Makefile | $(SED) -e "/^# EXPORT DELETE/d" > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile digestmd5/digestmd5.c gssapi/gssapi.c

# CRYPT DELETE END
# EXPORT DELETE END

include ../Makefile.targ

.PARALLEL: $(SUBDIRS)
```

```
**********************************************************
  149371 Thu Jul 11 01:29:37 2013
new/usr/src/lib/sasl_plugins/digestmd5/digestmd5.c
first pass
**********************************************************
```

```
   1 /*
   2  * Copyright 2003 Sun Microsystems, Inc.  All rights reserved.
   3  * Use is subject to license terms.
   4  */

   6 #pragma ident   "%Z%%M% %I%     %E% SMI"

   8 /* DIGEST-MD5 SASL plugin
   9  * Rob Siemborski
  10  * Tim Martin
  11  * Alexey Melnikov
  12  * $Id: digestmd5.c,v 1.153 2003/03/30 22:17:06 leg Exp $
  13  */
  14 /*
  15  * Copyright (c) 1998-2003 Carnegie Mellon University.  All rights reserved.
  16  *
  17  * Redistribution and use in source and binary forms, with or without
  18  * modification, are permitted provided that the following conditions
  19  * are met:
  20  *
  21  * 1. Redistributions of source code must retain the above copyright
  22  *    notice, this list of conditions and the following disclaimer.
  23  *
  24  * 2. Redistributions in binary form must reproduce the above copyright
  25  *    notice, this list of conditions and the following disclaimer in
  26  *    the documentation and/or other materials provided with the
  27  *    distribution.
  28  *
  29  * 3. The name "Carnegie Mellon University" must not be used to
  30  *    endorse or promote products derived from this software without
  31  *    prior written permission. For permission or any other legal
  32  *    details, please contact
  33  *      Office of Technology Transfer
  34  *      Carnegie Mellon University
  35  *      5000 Forbes Avenue
  36  *      Pittsburgh, PA  15213-3890
  37  *      (412) 268-4387, fax: (412) 268-7395
  38  *      tech-transfer@andrew.cmu.edu
  39  *
  40  * 4. Redistributions of any form whatsoever must retain the following
  41  *    acknowledgment:
  42  *    "This product includes software developed by Computing Services
  43  *     at Carnegie Mellon University (http://www.cmu.edu/computing/)."
  44  *
  45  * CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO
  46  * THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
  47  * AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE
  48  * FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
  49  * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN
  50  * AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING
  51  * OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
  52  */

  54 #include <config.h>

  56 #include <stdlib.h>
  57 #include <stdio.h>
  58 #include <string.h>
  59 #ifndef macintosh
  60 #include <sys/types.h>
  61 #include <sys/stat.h>
```



```
  62 #endif
  63 #include <fcntl.h>
  64 #include <ctype.h>

  66 /* EXPORT DELETE START */
  66 /* DES support */
  67 #ifdef WITH_DES
  68 # ifdef WITH_SSL_DES
  69 #  include <openssl/des.h>
  70 # else /* system DES library */
  71 #  include <des.h>
  72 # endif
  73 #endif /* WITH_DES */
  75 /* EXPORT DELETE END */

  75 #ifdef WIN32
  76 # include <winsock.h>
  77 #else /* Unix */
  78 # include <netinet/in.h>
  79 #endif /* WIN32 */

  81 #ifdef _SUN_SDK_
  82 #include <unistd.h>
  83 #endif /* _SUN_SDK_ */

  85 #include <sasl.h>
  86 #include <saslplug.h>

  88 #include "plugin_common.h"

  90 #if defined _SUN_SDK_  && defined USE_UEF
  91 #include <security/cryptoki.h>
  92 static int uef_init(const sasl_utils_t *utils);
  93 #endif /* _SUN_SDK_ && USE_UEF */

  95 #ifndef WIN32
  96 extern int strcasecmp(const char *s1, const char *s2);
  97 #endif /* end WIN32 */

  99 #ifdef macintosh
 100 #include <sasl_md5_plugin_decl.h>
 101 #endif

 103 /* external definitions */

 105 #ifndef _SUN_SDK_
 106 #ifdef sun
 107 /* gotta define gethostname ourselves on suns */
 108 extern int      gethostname(char *, int);
 109 #endif
 110 #endif /* !_SUN_SDK_ */

 112 #define bool int

 114 #ifndef TRUE
 115 #define TRUE  (1)
 116 #define FALSE (0)
 117 #endif

 119 #define DEFAULT_BUFSIZE 0xFFFF

 121 /*****************************  Common Section  *****************************/

 123 #ifndef _SUN_SDK_
 124 static const char plugin_id[] = "$Id: digestmd5.c,v 1.153 2003/03/30 22:17:06 le
 125 #endif /* !_SUN_SDK_ */
```

```
127 /* Definitions */
128 #define NONCE_SIZE (32)         /* arbitrary */

130 /* Layer Flags */
131 #define DIGEST_NOLAYER    (1)
132 #define DIGEST_INTEGRITY  (2)
133 #define DIGEST_PRIVACY    (4)

135 /* defines */
136 #define HASHLEN 16
137 typedef unsigned char HASH[HASHLEN + 1];
138 #define HASHHEXLEN 32
139 typedef unsigned char HASHHEX[HASHHEXLEN + 1];

141 #define MAC_SIZE 10
142 #define MAC_OFFS 2

144 const char *SEALING_CLIENT_SERVER="Digest H(A1) to client-to-server sealing key
145 const char *SEALING_SERVER_CLIENT="Digest H(A1) to server-to-client sealing key

147 const char *SIGNING_CLIENT_SERVER="Digest session key to client-to-server signin
148 const char *SIGNING_SERVER_CLIENT="Digest session key to server-to-client signin

150 #define HT      (9)
151 #define CR      (13)
152 #define LF      (10)
153 #define SP      (32)
154 #define DEL     (127)

156 struct context;

158 /* function definitions for cipher encode/decode */
159 typedef int cipher_function_t(struct context *,
160                               const char *,
161                               unsigned,
162                               unsigned char[],
163                               char *,
164                               unsigned *);

166 #ifdef _SUN_SDK_
167 typedef int cipher_init_t(struct context *, char [16],
168                                             char [16]);
169 #else
170 typedef int cipher_init_t(struct context *, unsigned char [16],
171                                             unsigned char [16]);
172 #endif /* _SUN_SDK_ */

174 typedef void cipher_free_t(struct context *);

176 enum Context_type { SERVER = 0, CLIENT = 1 };

178 typedef struct cipher_context cipher_context_t;

180 /* cached auth info used for fast reauth */
181 typedef struct reauth_entry {
182     char *authid;
183     char *realm;
184     unsigned char *nonce;
185     unsigned int nonce_count;
186     unsigned char *cnonce;

188     union {
189         struct {
190             time_t timestamp;
191         } s; /* server stuff */
```



```
193         struct {
194             char *serverFQDN;
195             int protection;
196             struct digest_cipher *cipher;
197             unsigned int server_maxbuf;
198         } c; /* client stuff */
199     } u;
200 } reauth_entry_t;
______unchanged_portion_omitted_

724 /* EXPORT DELETE START */
722 #ifdef WITH_DES
723 struct des_context_s {
724     des_key_schedule keysched;  /* key schedule for des initialization */
725     des_cblock ivec;            /* initial vector for encoding */
726     des_key_schedule keysched2; /* key schedule for 3des initialization */
727 };
______unchanged_portion_omitted_

1188 #endif /* WITH_RC4 */
1192 /* EXPORT DELETE END */

1190 struct digest_cipher available_ciphers[] =
1191 {
1196     /* EXPORT DELETE START */
1192 #ifdef WITH_RC4
1193     { "rc4-40", 40, 5, 0x01, &enc_rc4, &dec_rc4, &init_rc4, &free_rc4 },
1194     { "rc4-56", 56, 7, 0x02, &enc_rc4, &dec_rc4, &init_rc4, &free_rc4 },
1195     { "rc4", 128, 16, 0x04, &enc_rc4, &dec_rc4, &init_rc4, &free_rc4 },
1196 #endif
1197 #ifdef WITH_DES
1198     { "des", 55, 16, 0x08, &enc_des, &dec_des, &init_des, &free_des },
1199     { "3des", 112, 16, 0x10, &enc_3des, &dec_3des, &init_3des, &free_des },
1200 #endif
1206     /* EXPORT DELETE END */
1201     { NULL, 0, 0, 0, NULL, NULL, NULL, NULL }
1202 };
______unchanged_portion_omitted_

3681 static sasl_server_plug_t digestmd5_server_plugins[] =
3682 {
3683     {
3684         "DIGEST-MD5",                   /* mech_name */
3691         /* EXPORT DELETE START */
3685 #ifdef WITH_RC4
3686         128,                            /* max_ssf */
3687 #elif WITH_DES
3688         112,
3689 #else
3697         /* EXPORT DELETE END */
3690         0,
3699         /* EXPORT DELETE START */
3691 #endif
3701         /* EXPORT DELETE END */
3692         SASL_SEC_NOPLAINTEXT
3693         | SASL_SEC_NOANONYMOUS
3694         | SASL_SEC_MUTUAL_AUTH,         /* security_flags */
3695         SASL_FEAT_ALLOWS_PROXY,         /* features */
3696         NULL,                           /* glob_context */
3697         &digestmd5_server_mech_new,     /* mech_new */
3698         &digestmd5_server_mech_step,    /* mech_step */
3699         &digestmd5_server_mech_dispose, /* mech_dispose */
3700         &digestmd5_common_mech_free,    /* mech_free */
3701         NULL,                           /* setpass */
3702         NULL,                           /* user_query */
```

```
3703         NULL,                          /* idle */
3704         NULL,                          /* mech avail */
3705         NULL                           /* spare */
3706     }
3707 };

3709 int digestmd5_server_plug_init(sasl_utils_t *utils,
3710                                int maxversion,
3711                                int *out_version,
3712                                sasl_server_plug_t **pluglist,
3713                                int *plugcount)
3714 {
3715     reauth_cache_t *reauth_cache;
3716     const char *timeout = NULL;
3717     unsigned int len;
3718 #if defined _SUN_SDK_  && defined USE_UEF
3719     int ret;
3720 #endif /* _SUN_SDK_ && USE_UEF */

3722     if (maxversion < SASL_SERVER_PLUG_VERSION)
3723         return SASL_BADVERS;

3725 #if defined _SUN_SDK_  && defined USE_UEF
3726     if ((ret = uef_init(utils)) != SASL_OK)
3727         return ret;
3728 #endif /* _SUN_SDK_ && USE_UEF */

3730     /* reauth cache */
3731     reauth_cache = utils->malloc(sizeof(reauth_cache_t));
3732     if (reauth_cache == NULL)
3733         return SASL_NOMEM;
3734     memset(reauth_cache, 0, sizeof(reauth_cache_t));
3735     reauth_cache->i_am = SERVER;

3737     /* fetch and canonify the reauth_timeout */
3738     utils->getopt(utils->getopt_context, "DIGEST-MD5", "reauth_timeout",
3739                   &timeout, &len);
3740     if (timeout)
3741         reauth_cache->timeout = (time_t) 60 * strtol(timeout, NULL, 10);
3742 #ifdef _SUN_SDK_
3743     else
3744         reauth_cache->timeout = 0;
3745 #endif /* _SUN_SDK_ */
3746     if (reauth_cache->timeout < 0)
3747         reauth_cache->timeout = 0;

3749     if (reauth_cache->timeout) {
3750         /* mutex */
3751         reauth_cache->mutex = utils->mutex_alloc();
3752         if (!reauth_cache->mutex)
3753             return SASL_FAIL;

3755         /* entries */
3756         reauth_cache->size = 100;
3757         reauth_cache->e = utils->malloc(reauth_cache->size *
3758                                         sizeof(reauth_entry_t));
3759         if (reauth_cache->e == NULL)
3760             return SASL_NOMEM;
3761         memset(reauth_cache->e, 0, reauth_cache->size * sizeof(reauth_entry_t));
3762     }

3764     digestmd5_server_plugins[0].glob_context = reauth_cache;

3766 #ifdef _SUN_SDK_
3767 #ifdef USE_UEF_CLIENT
3768     digestmd5_server_plugins[0].max_ssf = uef_max_ssf;
```



```
3769 #endif /* USE_UEF_CLIENT */
3770 #endif /* _SUN_SDK_ */

3782     /* EXPORT DELETE START */
3783     /* CRYPT DELETE START */
3772 #ifdef _INTEGRATED_SOLARIS_
3773     /*
3774      * Let libsasl know that we are a "Sun" plugin so that privacy
3775      * and integrity will be allowed.
3776      */
3777     REG_PLUG("DIGEST-MD5", digestmd5_server_plugins);
3778 #endif /* _INTEGRATED_SOLARIS_ */
3791     /* CRYPT DELETE END */
3792     /* EXPORT DELETE END */

3780     *out_version = SASL_SERVER_PLUG_VERSION;
3781     *pluglist = digestmd5_server_plugins;
3782     *plugcount = 1;
3783
3784     return SASL_OK;
3785 }
______unchanged_portion_omitted_

5163 static sasl_client_plug_t digestmd5_client_plugins[] =
5164 {
5165     {
5166         "DIGEST-MD5",
5181         /* EXPORT DELETE START */
5167 #ifdef WITH_RC4                         /* mech_name */
5168         128,                            /* max ssf */
5169 #elif WITH_DES
5170         112,
5171 #else
5187         /* EXPORT DELETE END */
5172         0,
5189         /* EXPORT DELETE START */
5173 #endif
5191         /* EXPORT DELETE END */
5174         SASL_SEC_NOPLAINTEXT
5175         | SASL_SEC_NOANONYMOUS
5176         | SASL_SEC_MUTUAL_AUTH,         /* security_flags */
5177         SASL_FEAT_ALLOWS_PROXY,         /* features */
5178         NULL,                           /* required_prompts */
5179         NULL,                           /* glob_context */
5180         &digestmd5_client_mech_new,     /* mech_new */
5181         &digestmd5_client_mech_step,    /* mech_step */
5182         &digestmd5_client_mech_dispose, /* mech_dispose */
5183         &digestmd5_common_mech_free,    /* mech_free */
5184         NULL,                           /* idle */
5185         NULL,                           /* spare1 */
5186         NULL                            /* spare2 */
5187     }
5188 };

5190 int digestmd5_client_plug_init(sasl_utils_t *utils,
5191                                int maxversion,
5192                                int *out_version,
5193                                sasl_client_plug_t **pluglist,
5194                                int *plugcount)
5195 {
5196     reauth_cache_t *reauth_cache;
5197 #if defined _SUN_SDK_  && defined USE_UEF
5198     int ret;
5199 #endif /* _SUN_SDK_ && USE_UEF */

5201     if (maxversion < SASL_CLIENT_PLUG_VERSION)
```

```
5202         return SASL_BADVERS;
5203
5204 #if defined _SUN_SDK_  && defined USE_UEF
5205     if ((ret = uef_init(utils)) != SASL_OK)
5206         return ret;
5207 #endif /* _SUN_SDK_ && USE_UEF */

5209     /* reauth cache */
5210     reauth_cache = utils->malloc(sizeof(reauth_cache_t));
5211     if (reauth_cache == NULL)
5212         return SASL_NOMEM;
5213     memset(reauth_cache, 0, sizeof(reauth_cache_t));
5214     reauth_cache->i_am = CLIENT;
5215
5216     /* mutex */
5217     reauth_cache->mutex = utils->mutex_alloc();
5218     if (!reauth_cache->mutex)
5219         return SASL_FAIL;

5221     /* entries */
5222     reauth_cache->size = 10;
5223     reauth_cache->e = utils->malloc(reauth_cache->size *
5224                                     sizeof(reauth_entry_t));
5225     if (reauth_cache->e == NULL)
5226         return SASL_NOMEM;
5227     memset(reauth_cache->e, 0, reauth_cache->size * sizeof(reauth_entry_t));

5229     digestmd5_client_plugins[0].glob_context = reauth_cache;
5230 #ifdef _SUN_SDK_
5231 #ifdef USE_UEF_CLIENT
5232     digestmd5_client_plugins[0].max_ssf = uef_max_ssf;
5233 #endif /* USE_UEF_CLIENT */
5234 #endif /* _SUN_SDK_ */

5254     /* EXPORT DELETE START */
5255     /* CRYPT DELETE START */
5236 #ifdef _INTEGRATED_SOLARIS_
5237     /*
5238      * Let libsasl know that we are a "Sun" plugin so that privacy
5239      * and integrity will be allowed.
5240      */
5241     REG_PLUG("DIGEST-MD5", digestmd5_client_plugins);
5242 #endif /* _INTEGRATED_SOLARIS_ */
5263     /* CRYPT DELETE END */
5264     /* EXPORT DELETE END */

5244     *out_version = SASL_CLIENT_PLUG_VERSION;
5245     *pluglist = digestmd5_client_plugins;
5246     *plugcount = 1;
5247
5248     return SASL_OK;
5249 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   59831 Thu Jul 11 01:29:38 2013
new/usr/src/lib/sasl_plugins/gssapi/gssapi.c
first pass
**********************************************************
______unchanged_portion_omitted_

1439 int gssapiv2_server_plug_init(
1440 #ifndef HAVE_GSSKRB5_REGISTER_ACCEPTOR_IDENTITY
1441     const sasl_utils_t *utils __attribute__((unused)),
1442 #else
1443     const sasl_utils_t *utils,
1444 #endif
1445     int maxversion,
1446     int *out_version,
1447     sasl_server_plug_t **pluglist,
1448     int *plugcount)
1449 {
1450 #ifdef HAVE_GSSKRB5_REGISTER_ACCEPTOR_IDENTITY
1451     const char *keytab = NULL;
1452     char keytab_path[1024];
1453     unsigned int rl;
1454 #endif
1455
1456     if (maxversion < SASL_SERVER_PLUG_VERSION) {
1457         return SASL_BADVERS;
1458     }
1459
1460 #ifndef _SUN_SDK_
1461 #ifdef HAVE_GSSKRB5_REGISTER_ACCEPTOR_IDENTITY
1462     /* unfortunately, we don't check for readability of keytab if it's
1463        the standard one, since we don't know where it is */
1464
1465     /* FIXME: This code is broken */
1466
1467     utils->getopt(utils->getopt_context, "GSSAPI", "keytab", &keytab, &rl);
1468     if (keytab != NULL) {
1469         if (access(keytab, R_OK) != 0) {
1470             utils->log(NULL, SASL_LOG_ERR,
1471                        "Could not find keytab file: %s: %m",
1472                        keytab, errno);
1473             return SASL_FAIL;
1474         }
1475
1476         if(strlen(keytab) > 1024) {
1477             utils->log(NULL, SASL_LOG_ERR,
1478                        "path to keytab is > 1024 characters");
1479             return SASL_BUFOVER;
1480         }
1481
1482         strncpy(keytab_path, keytab, 1024);
1483
1484         gsskrb5_register_acceptor_identity(keytab_path);
1485     }
1486 #endif
1487 #endif /* !_SUN_SDK_ */
1488
1489     /* EXPORT DELETE START */
1490     /* CRYPT DELETE START */
1489 #ifdef _INTEGRATED_SOLARIS_
1490     /*
1491      * Let libsasl know that we are a "Sun" plugin so that privacy
1492      * and integrity will be allowed.
1493      */
1494     REG_PLUG("GSSAPI", gssapi_server_plugins);
1495 #endif /* _INTEGRATED_SOLARIS_ */
```



```
1498     /* CRYPT DELETE END */
1499     /* EXPORT DELETE END */

1497     *out_version = SASL_SERVER_PLUG_VERSION;
1498     *pluglist = gssapi_server_plugins;
1499     *plugcount = 1;
1500
1501     return SASL_OK;
1502 }
______unchanged_portion_omitted_

2174 int gssapiv2_client_plug_init(const sasl_utils_t *utils __attribute__((unused)),
2175                                int maxversion,
2176                                int *out_version,
2177                                sasl_client_plug_t **pluglist,
2178                                int *plugcount)
2179 {
2180     if (maxversion < SASL_CLIENT_PLUG_VERSION) {
2181         SETERROR(utils, "Version mismatch in GSSAPI");
2182         return SASL_BADVERS;
2183     }
2184
2189     /* EXPORT DELETE START */
2190     /* CRYPT DELETE START */
2185 #ifdef _INTEGRATED_SOLARIS_
2186     /*
2187      * Let libsasl know that we are a "Sun" plugin so that privacy
2188      * and integrity will be allowed.
2189      */
2190     REG_PLUG("GSSAPI", gssapi_client_plugins);
2191 #endif /* _INTEGRATED_SOLARIS_ */
2198     /* CRYPT DELETE END */
2199     /* EXPORT DELETE END */

2193     *out_version = SASL_CLIENT_PLUG_VERSION;
2194     *pluglist = gssapi_client_plugins;
2195     *plugcount = 1;
2196
2197     return SASL_OK;
2198 }
______unchanged_portion_omitted_
```

```
**********************************************************
   24459 Thu Jul 11 01:29:39 2013
new/usr/src/pkg/Makefile
first pass
**********************************************************
```

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

  26 include $(SRC)/Makefile.master
  27 include $(SRC)/Makefile.buildnum

  29 #
  30 # Make sure we're getting a consistent execution environment for the
  31 # embedded scripts.
  32 #
  33 SHELL= /usr/bin/ksh93

  35 #
  36 # To suppress package dependency generation on any system, regardless
  37 # of how it was installed, set SUPPRESSPKGDEP=true in the build
  38 # environment.
  39 #
  40 SUPPRESSPKGDEP= false

  42 #
  43 # Comment this line out or set "PKGDEBUG=" in your build environment
  44 # to get more verbose output from the make processes in usr/src/pkg
  45 #
  46 PKGDEBUG= @

  48 #
  49 # Cross platform packaging notes
  50 #
  51 # By default, we package the proto area from the same architecture as
  52 # the packaging build.  In other words, if you're running nightly or
  53 # bldenv on an x86 platform, it will take objects from the x86 proto
  54 # area and use them to create x86 repositories.
  55 #
  56 # If you want to create repositories for an architecture that's
  57 # different from $(uname -p), you do so by setting PKGMACH in your
  58 # build environment.
  59 #
  60 # For this to work correctly, the following must all happen:
  61 #
```



```
  62 #   1. You need the desired proto area, which you can get either by
  63 #      doing a gatekeeper-style build with the -U option to
  64 #      nightly(1), or by using rsync.  If you don't do this, you will
  65 #      get packaging failures building all packages, because pkgsend
  66 #      is unable to find the required binaries.
  67 #   2. You need the desired tools proto area, which you can get in the
  68 #      same ways as the normal proto area.  If you don't do this, you
  69 #      will get packaging failures building onbld, because pkgsend is
  70 #      unable to find the tools binaries.
  71 #   3. The remainder of this Makefile should never refer directly to
  72 #      $(MACH).  Instead, $(PKGMACH) should be used whenever an
  73 #      architecture-specific path or token is needed.  If this is done
  74 #      incorrectly, then packaging will fail, and you will see the
  75 #      value of $(uname -p) instead of the value of $(PKGMACH) in the
  76 #      commands that fail.
  77 #   4. Each time a rule in this Makefile invokes $(MAKE), it should
  78 #      pass PKGMACH=$(PKGMACH) explicitly on the command line.  If
  79 #      this is done incorrectly, then packaging will fail, and you
  80 #      will see the value of $(uname -p) instead of the value of
  81 #      $(PKGMACH) in the commands that fail.
  82 #
  83 # Refer also to the convenience targets defined later in this
  84 # Makefile.
  85 #
  86 PKGMACH=        $(MACH)

  88 #
  89 # ROOT, TOOLS_PROTO, and PKGARCHIVE should be set by nightly or
  90 # bldenv.  These macros translate them into terms of $PKGMACH, instead
  91 # of $ARCH.
  92 #
  93 PKGROOT.cmd=    print $(ROOT) | sed -e s:/root_$(MACH):/root_$(PKGMACH):
  94 PKGROOT=        $(PKGROOT.cmd:sh)
  95 TOOLSROOT.cmd=  print $(TOOLS_PROTO) | sed -e s:/root_$(MACH):/root_$(PKGMACH):
  96 TOOLSROOT=      $(TOOLSROOT.cmd:sh)
  97 PKGDEST.cmd=    print $(PKGARCHIVE) | sed -e s:/$(MACH)/:/$(PKGMACH)/:
  98 PKGDEST=        $(PKGDEST.cmd:sh)

 100 EXCEPTIONS= packaging

 102 PKGMOGRIFY= pkgmogrify

 104 #
 105 # Always build the redistributable repository, but only build the
 106 # nonredistributable bits if we have access to closed source.
 107 #
 108 # Some objects that result from the closed build are still
 109 # redistributable, and should be packaged as part of an open-only
 110 # build.  Access to those objects is provided via the closed-bins
 111 # tarball.  See usr/src/tools/scripts/bindrop.sh for details.
 112 #
 113 REPOS= redist

 115 #
 116 # The packages directory will contain the processed manifests as
 117 # direct build targets and subdirectories for package metadata extracted
 118 # incidentally during manifest processing.
 119 #
 120 # Nothing underneath $(PDIR) should ever be managed by SCM.
 121 #
 122 PDIR= packages.$(PKGMACH)

 124 #
 125 # The tools proto must be specified for dependency generation.
 126 # Publication from the tools proto area is managed in the
 127 # publication rule.
```

```
128 #
129 $(PDIR)/developer-build-onbld.dep:= PKGROOT= $(TOOLSROOT)

131 PKGPUBLISHER= $(PKGPUBLISHER_REDIST)

133 #
134 # To get these defaults, manifests should simply refer to $(PKGVERS).
135 #
136 PKGVERS_COMPONENT= 0.$(RELEASE)
137 PKGVERS_BUILTON= $(RELEASE)
138 PKGVERS_BRANCH= 0.$(ONNV_BUILDNUM)
139 PKGVERS= $(PKGVERS_COMPONENT),$(PKGVERS_BUILTON)-$(PKGVERS_BRANCH)

141 #
142 # The ARCH32 and ARCH64 macros are used in the manifests to express
143 # architecture-specific subdirectories in the installation paths
144 # for isaexec'd commands.
145 #
146 # We can't simply use $(MACH32) and $(MACH64) here, because they're
147 # only defined for the build architecture.  To do cross-platform
148 # packaging, we need both values.
149 #
150 i386_ARCH32= i86
151 sparc_ARCH32= sparcv7
152 i386_ARCH64= amd64
153 sparc_ARCH64= sparcv9

155 OPENSSL =       /usr/bin/openssl
156 OPENSSL10.cmd = $(OPENSSL) version | $(NAWK) '{if($$2<1){print "\043";}}'
157 OPENSSL10_ONLY  =       $(OPENSSL10.cmd:sh)

159 #
160 # macros and transforms needed by pkgmogrify
161 #
162 # If you append to this list using target-specific assignments (:=),
163 # be very careful that the targets are of the form $(PDIR)/pkgname.  If
164 # you use a higher level target, or a package list, you'll trigger a
165 # complete reprocessing of all manifests because they'll fail command
166 # dependency checking.
167 #
168 PM_TRANSFORMS= common_actions publish restart_fmri facets defaults \
169         extract_metadata
170 PM_INC= transforms manifests

172 PKGMOG_DEFINES= \
173         i386_ONLY=$(POUND_SIGN) \
174         sparc_ONLY=$(POUND_SIGN) \
175         OPENSSL10_ONLY=$(OPENSSL10_ONLY) \
176         $(PKGMACH)_ONLY= \
177         ARCH=$(PKGMACH) \
178         ARCH32=$($(PKGMACH)_ARCH32) \
179         ARCH64=$($(PKGMACH)_ARCH64) \
180         PKGVERS_COMPONENT=$(PKGVERS_COMPONENT) \
181         PKGVERS_BUILTON=$(PKGVERS_BUILTON) \
182         PKGVERS_BRANCH=$(PKGVERS_BRANCH) \
183         PKGVERS=$(PKGVERS)

185 PKGDEP_TOKENS_i386= \
186         'PLATFORM=i86hvm' \
187         'PLATFORM=i86pc' \
188         'PLATFORM=i86xpv' \
189         'ISALIST=amd64' \
190         'ISALIST=i386'
191 PKGDEP_TOKENS_sparc= \
192         'PLATFORM=sun4u' \
193         'PLATFORM=sun4v' \
```



```
194         'ISALIST=sparcv9' \
195         'ISALIST=sparc'
196 PKGDEP_TOKENS= $(PKGDEP_TOKENS_$(PKGMACH))

198 #
199 # The package lists are generated with $(PKGDEP_TYPE) as their
200 # dependency types, so that they can be included by either an
201 # incorporation or a group package.
202 #
203 $(PDIR)/osnet-redist.mog := PKGDEP_TYPE= require
204 $(PDIR)/osnet-incorporation.mog:= PKGDEP_TYPE= incorporate

206 PKGDEP_INCORP= \
207         depend fmri=consolidation/osnet/osnet-incorporation type=require

209 #
210 # All packaging build products should go into $(PDIR), so they don't
211 # need to be included separately in CLOBBERFILES.
212 #
213 CLOBBERFILES= $(PDIR) proto_list_$(PKGMACH)

215 #
216 # By default, PKGS will list all manifests.  To build and/or publish a
217 # subset of packages, override this on the command line or in the
218 # build environment and then reference (implicitly or explicitly) the all
219 # or install targets.
220 #
221 MANIFESTS :sh= (cd manifests; print *.mf)
222 PKGS= $(MANIFESTS:%.mf=%)
223 DEP_PKGS= $(PKGS:%=$(PDIR)/%.dep)
224 PROC_PKGS= $(PKGS:%=$(PDIR)/%.mog)

226 #
227 # Track the synthetic manifests separately so we can properly express
228 # build rules and dependencies.  The synthetic and real packages use
229 # different sets of transforms and macros for pkgmogrify.
230 #
231 SYNTH_PKGS= osnet-incorporation osnet-redist
232 DEP_SYNTH_PKGS= $(SYNTH_PKGS:%=$(PDIR)/%.dep)
233 PROC_SYNTH_PKGS= $(SYNTH_PKGS:%=$(PDIR)/%.mog)

235 #
236 # Root of pkg image to use for dependency resolution
237 # Normally / on the machine used to build the binaries
238 #
239 PKGDEP_RESOLVE_IMAGE = /

241 #
242 # For each package, we determine the target repository based on
243 # manifest-embedded metadata.  Because we make that determination on
244 # the fly, the publication target cannot be expressed as a
245 # subdirectory inside the unknown-by-the-makefile target repository.
246 #
247 # In order to limit the target set to real files in known locations,
248 # we use a ".pub" file in $(PDIR) for each processed manifest, regardless
249 # of content or target repository.
250 #
251 PUB_PKGS= $(SYNTH_PKGS:%=$(PDIR)/%.pub) $(PKGS:%=$(PDIR)/%.pub)

253 #
254 # Any given repository- and status-specific package list may be empty,
255 # but we can only determine that dynamically, so we always generate all
256 # lists for each repository we're building.
257 #
258 # The meanings of each package status are as follows:
259 #
```

```
260 #       PKGSTAT         meaning
261 #       ----------      ------------------------------------------------------
262 #       noincorp        Do not include in incorporation or group package
263 #       obsolete        Include in incorporation, but not group package
264 #       renamed         Include in incorporation, but not group package
265 #       current         Include in incorporation and group package
266 #
267 # Since the semantics of the "noincorp" package status dictate that
268 # such packages are not included in the incorporation or group packages,
269 # there is no need to build noincorp package lists.
270 #
271 PKGLISTS= \
272         $(REPOS:%=$(PDIR)/packages.%.current) \
273         $(REPOS:%=$(PDIR)/packages.%.renamed) \
274         $(REPOS:%=$(PDIR)/packages.%.obsolete)

276 .KEEP_STATE:

278 .PARALLEL: $(PKGS) $(PROC_PKGS) $(DEP_PKGS) \
279         $(PROC_SYNTH_PKGS) $(DEP_SYNTH_PKGS) $(PUB_PKGS)

281 #
282 # For a single manifest, the dependency chain looks like this:
283 #
284 #       raw manifest (mypkg.mf)
285 #               |
286 #               | use pkgmogrify to process raw manifest
287 #               |
288 #       processed manifest (mypkg.mog)
289 #               |
290 #          *    | use pkgdepend generate to generate dependencies
291 #               |
292 #       manifest with TBD dependencies (mypkg.dep)
293 #               |
294 #          %    | use pkgdepend resolve to resolve dependencies
295 #               |
296 #       manifest with dependencies resolved (mypkg.res)
297 #               |
298 #               | use pkgsend to publish the package
299 #               |
300 #       placeholder to indicate successful publication (mypkg.pub)
301 #
302 # * This may be suppressed via SUPPRESSPKGDEP.  The resulting
303 #   packages will install correctly, but care must be taken to
304 #   install all dependencies, because pkg will not have the input
305 #   it needs to determine this automatically.
306 #
307 # % This is included in this diagram to make the picture complete, but
308 #   this is a point of synchronization in the build process.
309 #   Dependency resolution is actually done once on the entire set of
310 #   manifests, not on a per-package basis.
311 #
312 # The full dependency chain for generating everything that needs to be
313 # published, without actually publishing it, looks like this:
314 #
315 #       processed synthetic packages
316 #               |               |
317 #       package lists       synthetic package manifests
318 #               |
319 #       processed real packages
320 #           |           |
321 #       package dir     real package manifests
322 #
323 # Here, each item is a set of real or synthetic packages.  For this
324 # portion of the build, no reference is made to the proto area.  It is
325 # therefore suitable for the "all" target, as opposed to "install."
```



```
326 #
327 # Since each of these steps is expressed explicitly, "all" need only
328 # depend on the head of the chain.
329 #
330 # From the end of manifest processing, the publication dependency
331 # chain looks like this:
332 #
333 #               repository metadata (catalogs and search indices)
334 #                       |
335 #                       | pkg.depotd
336 #                       |
337 #               published packages
338 #                 |               |
339 #                 |               | pkgsend publish
340 #                 |               |
341 #         repositories      resolved dependencies
342 #                 |                       |
343 # pkgsend         |                       | pkgdepend resolve
344 # create-repository                       |
345 #                 |               generated dependencies
346 #        repo directories                 |
347 #                                         | pkgdepend
348 #                                         |
349 #                                 processed manifests
350 #

352 ALL_TARGETS= $(PROC_SYNTH_PKGS) proto_list_$(PKGMACH)

354 all: $(ALL_TARGETS)

356 #
357 # This will build the directory to contain the processed manifests
358 # and the metadata symlinks.
359 #
360 $(PDIR):
361         @print "Creating $(@)"
362         $(PKGDEBUG)$(INS.dir)

364 #
365 # This rule resolves dependencies across all published manifests.
366 #
367 # We shouldn’t have to ignore the error from pkgdepend, but until
368 # 16012 and its dependencies are resolved, pkgdepend will always exit
369 # with an error.
370 #
371 $(PDIR)/gendeps: $(DEP_SYNTH_PKGS) $(DEP_PKGS)
372         -$(PKGDEBUG)if [ "$(SUPPRESSPKGDEP)" = "true" ]; then \
373                 print "Suppressing dependency resolution"; \
374                 for p in $(DEP_PKGS:%.dep=%); do \
375                         $(CP) $$p.dep $$p.res; \
376                 done; \
377         else \
378                 print "Resolving dependencies"; \
379                 pkgdepend -R $(PKGDEP_RESOLVE_IMAGE) resolve \
380                     -m $(DEP_SYNTH_PKGS) $(DEP_PKGS); \
381                 for p in $(DEP_SYNTH_PKGS:%.dep=%) $(DEP_PKGS:%.dep=%); do \
382                         if [ "$$(print $$p.metadata.*)" = \
383                              "$$(print $$p.metadata.noincorp.*)" ]; \
384                         then \
385                                 print "Removing dependency versions from $$p"; \
386                                 $(PKGMOGRIFY) $(PKGMOG_VERBOSE) \
387                                     -O $$p.res -I transforms \
388                                     strip_versions $$p.dep.res; \
389                                 $(RM) $$p.dep.res; \
390                         else \
391                                 $(MV) $$p.dep.res $$p.res; \
```

```
392                         fi; \
393                 done; \
394         fi
395         $(PKGDEBUG)$(TOUCH) $(@)

397 install: $(ALL_TARGETS) repository-metadata

399 repository-metadata: publish_pkgs
400         @print "Creating repository metadata"
401         $(PKGDEBUG)for r in $(REPOS); do \
402                 /usr/lib/pkg.depotd -d $(PKGDEST)/repo.$$r \
403                         --add-content --exit-ready; \
404         done

406 #
407 # Since we create zero-length processed manifests for a graceful abort
408 # from pkgmogrify, we need to detect that here and make no effort to
409 # publish the package.
410 #
411 # For all other packages, we publish them regardless of status.  We
412 # derive the target repository as a component of the metadata-derived
413 # symlink for each package.
414 #
415 publish_pkgs: $(REPOS:%=$(PKGDEST)/repo.%) $(PDIR)/gendeps .WAIT $(PUB_PKGS)

417 #
418 # Before publishing, we want to pull the license files from $CODEMGR_WS
419 # into the proto area.  This allows us to NOT pass $SRC (or
420 # $CODEMGR_WS) as a basedir for publication.
421 #
422 $(PUB_PKGS): stage-licenses

424 #
425 # Initialize the empty on-disk repositories
426 #
427 $(REPOS:%=$(PKGDEST)/repo.%):
428         @print "Initializing $(@F)"
429         $(PKGDEBUG)$(INS.dir)
430         $(PKGDEBUG)pkgsend -s file://$(@) create-repository \
431                 --set-property publisher.prefix=$(PKGPUBLISHER)

433 #
434 # rule to process real manifests
435 #
436 # To allow redistributability and package status to change, we must
437 # remove not only the actual build target (the processed manifest), but
438 # also the incidental ones (the metadata-derived symlinks).
439 #
440 # If pkgmogrify exits cleanly but fails to create the specified output
441 # file, it means that it encountered an abort directive.  That means
442 # that this package should not be published for this particular build
443 # environment.  Since we can't prune such packages from $(PKGS)
444 # retroactively, we need to create an empty target file to keep make
445 # from trying to rebuild it every time.  For these empty targets, we
446 # do not create metadata symlinks.
447 #
448 # Automatic dependency resolution to files is also done at this phase of
449 # processing.  The skipped packages are skipped due to existing bugs
450 # in pkgdepend.
451 #
452 # The incorporation dependency is tricky: it needs to go into all
453 # current and renamed manifests (ie all incorporated packages), but we
454 # don't know which those are until after we run pkgmogrify.  So
455 # instead of expressing it as a transform, we tack it on ex post facto.
456 #
457 # Implementation notes:
```



```
458 #
459 # - The first $(RM) must not match other manifests, or we'll run into
460 #   race conditions with parallel manifest processing.
461 #
462 # - The make macros [ie $(MACRO)] are evaluated when the makefile is
463 #   read in, and will result in a fixed, macro-expanded rule for each
464 #   target enumerated in $(PROC_PKGS).
465 #
466 # - The shell variables (ie $$VAR) are assigned on the fly, as the rule
467 #   is executed.  The results may only be referenced in the shell in
468 #   which they are assigned, so from the perspective of make, all code
469 #   that needs these variables needs to be part of the same line of
470 #   code.  Hence the use of command separators and line continuation
471 #   characters.
472 #
473 # - The extract_metadata transforms are designed to spit out shell
474 #   variable assignments to stdout.  Those are published to the
475 #   .vars temporary files, and then used as input to the eval
476 #   statement.  This is done in stages specifically so that pkgmogrify
477 #   can signal failure if the manifest has a syntactic or other error.
478 #   The eval statement should begin with the default values, and the
479 #   output from pkgmogrify (if any) should be in the form of a
480 #   variable assignment to override those defaults.
481 #
482 # - When this rule completes execution, it must leave an updated
483 #   target file ($@) in place, or make will reprocess the package
484 #   every time it encounters it as a dependency.  Hence the "touch"
485 #   statement to ensure that the target is created, even when
486 #   pkgmogrify encounters an abort in the publish transforms.
487 #

489 .SUFFIXES: .mf .mog .dep .res .pub

491 $(PDIR)/%.mog: manifests/%.mf
492         @print "Processing manifest $(<F)"
493         @env PKGFMT_OUTPUT=v1 pkgfmt -c $<
494         $(PKGDEBUG)$(RM) $(@) $(@:%.mog=%) $(@:%.mog=%.nodepend) \
495                 $(@:%.mog=%.lics) $(PDIR)/$(@F:%.mog=%).metadata.* $(@).vars
496         $(PKGDEBUG)$(PKGMOGRIFY) $(PKGMOG_VERBOSE) $(PM_INC:%= -I %) \
497                 $(PKGMOG_DEFINES:%=-D %) -P $(@).vars -O $(@) \
498                 $(<) $(PM_TRANSFORMS)
499         $(PKGDEBUG)eval REPO=redist PKGSTAT=current NODEPEND=$(SUPPRESSPKGDEP) \
500                 '$(CAT) -s $(@).vars'; \
501         if [ -f $(@) ]; then \
502                 if [ "$$NODEPEND" != "false" ]; then \
503                         $(TOUCH) $(@:%.mog=%.nodepend); \
504                 fi; \
505                 $(LN) -s $(@F) \
506                         $(PDIR)/$(@F:%.mog=%).metadata.$$PKGSTAT.$$REPO; \
507                 if [ \( "$$PKGSTAT" = "current" \) -o \
508                      \( "$$PKGSTAT" = "renamed" \) ]; \
509                         then print $(PKGDEP_INCORP) >> $(@); \
510                 fi; \
511                 print $$LICS > $(@:%.mog=%.lics); \
512         else \
513                 $(TOUCH) $(@) $(@:%.mog=%.lics); \
514         fi
515         $(PKGDEBUG)$(RM) $(@).vars

517 $(PDIR)/%.dep: $(PDIR)/%.mog
518         @print "Generating dependencies for $(<F)"
519         $(PKGDEBUG)$(RM) $(@)
520         $(PKGDEBUG)if [ ! -f $(@:%.dep=%.nodepend) ]; then \
521                 pkgdepend generate -m $(PKGDEP_TOKENS:%=-D %) $(<) \
522                         $(PKGROOT) > $(@); \
523         else \
```

```
524                 $(CP) $(<) $(@); \
525         fi

527 #
528 # The full chain implies that there should be a .dep.res suffix rule,
529 # but dependency generation is done on a set of manifests, rather than
530 # on a per-manifest basis.  Instead, see the gendeps rule above.
531 #

533 $(PDIR)/%.pub: $(PDIR)/%.res
534         $(PKGDEBUG)m=$$(basename $(@:%.pub=%).metadata.*); \
535         r=$${m#$(@F:%.pub=%.metadata.)+(?).}; \
536         if [ -s $(<) ]; then \
537                 print "Publishing $(@F:%.pub=%) to $$r repository"; \
538                 pkgsend -s file://$(PKGDEST)/repo.$$r publish \
539                     -d $(PKGROOT) -d $(TOOLSROOT) \
540                     -d license_files -d $(PKGROOT)/licenses \
541                     --fmri-in-manifest --no-index --no-catalog $(<) \
542                     > /dev/null; \
543         fi; \
544         $(TOUCH) $(@);

546 #
547 # rule to build the synthetic manifests
548 #
549 # This rule necessarily has PKGDEP_TYPE that changes according to
550 # the specific synthetic manifest.  Rather than escape command
551 # dependency checking for the real manifest processing, or failing to
552 # express the (indirect) dependency of synthetic manifests on real
553 # manifests, we simply split this rule out from the one above.
554 #
555 # The implementation notes from the previous rule are applicable
556 # here, too.
557 #
558 $(PROC_SYNTH_PKGS): $(PKGLISTS) $$(@F:%.mog=%.mf)
559         @print "Processing synthetic manifest $(@F:%.mog=%.mf)"
560         $(PKGDEBUG)$(RM) $(@) $(PDIR)/$(@F:%.mog=%).metadata.* $(@).vars
561         $(PKGDEBUG)$(PKGMOGRIFY) $(PKGMOG_VERBOSE) -I transforms -I $(PDIR) \
562                 $(PKGMOG_DEFINES:%=-D %) -D PKGDEP_TYPE=$(PKGDEP_TYPE) \
563                 -P $(@).vars -O $(@) $(@F:%.mog=%.mf) \
564                 $(PM_TRANSFORMS) synthetic
565         $(PKGDEBUG)eval REPO=redist PKGSTAT=current '$(CAT) -s $(@).vars'; \
566         if [ -f $(@) ]; then \
567                 $(LN) -s $(@F) \
568                         $(PDIR)/$(@F:%.mog=%).metadata.$$PKGSTAT.$$REPO; \
569         else \
570                 $(TOUCH) $(@); \
571         fi
572         $(PKGDEBUG)$(RM) $(@).vars

574 $(DEP_SYNTH_PKGS): $$(@:%.dep=%.mog)
575         @print "Skipping dependency generation for $(@F:%.dep=%)"
576         $(PKGDEBUG)$(CP) $(@:%.dep=%.mog) $(@)

578 clean:

580 clobber: clean
581         $(RM) -r $(CLOBBERFILES)

583 #
584 # This rule assumes that all links in the $PKGSTAT directories
585 # point to valid manifests, and will fail the make run if one
586 # does not contain an fmri.
587 #
588 # We do this in the BEGIN action instead of using pattern matching
589 # because we expect the fmri to be at or near the first line of each input
```



```
590 # file, and this way lets us avoid reading the rest of the file after we
591 # find what we need.
592 #
593 # We keep track of a failure to locate an fmri, so we can fail the
594 # make run, but we still attempt to process each package in the
595 # repo/pkgstat-specific subdir, in hopes of maybe giving some
596 # additional useful info.
597 #
598 # The protolist is used for bfu archive creation, which may be invoked
599 # interactively by the user.  Both protolist and PKGLISTS targets
600 # depend on $(PROC_PKGS), but protolist builds them recursively.
601 # To avoid collisions, we insert protolist into the dependency chain
602 # here.  This has two somewhat subtle benefits: it allows bfu archive
603 # creation to work correctly, even when -a was not part of NIGHTLY_OPTIONS,
604 # and it ensures that a protolist file here will always correspond to the
605 # contents of the processed manifests, which can vary depending on build
606 # environment.
607 #
608 $(PKGLISTS): $(PROC_PKGS)
609         $(PKGDEBUG)sdotr=$(@F:packages.%=%); \
610         r=$${sdotr%.+(?)}; s=$${sdotr#+(?).}; \
611         print "Generating $$r $$s package list"; \
612         $(RM) $(@); $(TOUCH) $(@); \
613         $(NAWK) 'BEGIN { \
614                 if (ARGC < 2) { \
615                         exit; \
616                 } \
617                 retcode = 0; \
618                 for (i = 1; i < ARGC; i++) { \
619                         do { \
620                                 e = getline f < ARGV[i]; \
621                         } while ((e == 1) && (f !~ /name=pkg.fmri/)); \
622                         close(ARGV[i]); \
623                         if (e == 1) { \
624                                 l = split(f, a, "="); \
625                                 print "depend fmri=" a[l], \
626                                         "type=$$(PKGDEP_TYPE)"; \
627                         } else { \
628                                 print "no fmri in " ARGV[i] >> "/dev/stderr"; \
629                                 retcode = 2; \
630                         } \
631                 } \
632                 exit retcode; \
633         }' 'find $(PDIR) -type l -a \( $(PKGS:%=-name %.metadata.$$s.$$r -o) \
634                 -name NOSUCHFILE \)' >> $(@)

636 #
637 # rules to validate proto area against manifests, check for safe
638 # file permission modes, and generate a faux proto list
639 #
640 # For the check targets, the dependencies on $(PROC_PKGS) is specified
641 # as a subordinate make process in order to suppress output.
642 #
643 makesilent:
644         @$(MAKE) -e $(PROC_PKGS) PKGMACH=$(PKGMACH) \
645                 SUPPRESSPKGDEP=$(SUPPRESSPKGDEP) > /dev/null

647 #
648 # The .lics files were created during pkgmogrification, and list the
649 # set of licenses to pull from $SRC for each package.  Because
650 # licenses may be duplicated between packages, we uniquify them as
651 # well as aggregating them here.
652 #
653 license-list: makesilent
654         $(PKGDEBUG)( for l in 'cat $(PROC_PKGS:%.mog=%.lics)'; \
655                 do print $$l; done ) | sort -u > $@
```

```
657 #
658 # Staging the license and description files in the proto area allows
659 # us to do proper unreferenced file checking of both license and
660 # description files without blanket exceptions, and to pull license
661 # content without reference to $CODEMGR_WS during publication.
662 #
663 stage-licenses: license-list FRC
664         $(PKGDEBUG)$(MAKE) -e -f Makefile.lic \
665                 PKGDEBUG=$(PKGDEBUG) LICROOT=$(PKGROOT)/licenses \
666                 '$(NAWK) '{ \
667                         print "$(PKGROOT)/licenses/" $$0; \
668                         print "$(PKGROOT)/licenses/" $$0 ".descrip"; \
669                 }' license-list' > /dev/null;

671 protocmp: makesilent
672         @validate_pkg -a $(PKGMACH) -v \
673                 $(EXCEPTIONS:%=-e $(CODEMGR_WS)/exception_lists/%) \
674                 -m $(PDIR) -p $(PKGROOT) -p $(TOOLSROOT)

676 pmodes: makesilent
677         @validate_pkg -a $(PKGMACH) -M -m $(PDIR) \
678                 -e $(CODEMGR_WS)/exception_lists/pmodes

680 check: protocmp pmodes

682 protolist: proto_list_$(PKGMACH)

684 proto_list_$(PKGMACH): $(PROC_PKGS)
685         @validate_pkg -a $(PKGMACH) -L -m $(PDIR) > $(@)

687 $(PROC_PKGS): $(PDIR)

689 #
690 # This is a convenience target to allow package names to function as
691 # build targets.  Generally, using it is only useful when iterating on
692 # development of a manifest.
693 #
694 # When processing a manifest, use the basename (without extension) of
695 # the package.  When publishing, use the basename with a ".pub"
696 # extension.
697 #
698 # Other than during manifest development, the preferred usage is to
699 # avoid these targets and override PKGS on the make command line and
700 # use the provided all and install targets.
701 #
702 $(PKGS) $(SYNTH_PKGS): $(PDIR)/$$(@:%=%.mog)

704 $(PKGS:%=%.pub) $(SYNTH_PKGS:%=%.pub): $(PDIR)/$$(@)

706 #
707 # This is a convenience target to resolve dependencies without publishing
708 # packages.
709 #
710 gendeps: $(PDIR)/gendeps

712 #
713 # These are convenience targets for cross-platform packaging.  If you
714 # want to build any of "the normal" targets for a different
715 # architecture, simply use "arch/target" as your build target.
716 #
717 # Since the most common use case for this is "install," the architecture
718 # specific install targets have been further abbreviated to elide "/install."
719 #
720 i386/% sparc/%:
721         $(MAKE) -e $(@F) PKGMACH=$(@D) SUPPRESSPKGDEP=$(SUPPRESSPKGDEP)
```



```
723 i386 sparc: $$(@)/install

725 FRC:

727 # EXPORT DELETE START
728 XMOD_PKGS= \
729         BRCMbnx \
730         BRCMbnxe \
731         SUNWadpu320 \
732         SUNWibsdpib \
733         SUNWkdc \
734         SUNWlsimega \
735         SUNWwbint \
736         SUNWwbsup

738 EXPORT_SRC: CRYPT_SRC
739         $(RM) $(XMOD_PKGS:%=manifests/%.mf)
740         $(RM) Makefile+
741         $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
742                 < Makefile > Makefile+
743         $(MV) -f Makefile+ Makefile
744         $(CHMOD) 444 Makefile
745 # EXPORT DELETE END
```

```
**********************************************************
    2084 Thu Jul 11 01:29:40 2013
new/usr/src/psm/stand/boot/Makefile
first pass
**********************************************************
```

```
#
# 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 2010 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

include         ../../../Makefile.master

sparcv9_ARCHITECTURES = sparcv9
sparc_ARCHITECTURES = $(sparcv9_ARCHITECTURES)

SUBDIRS = $($(MACH)_ARCHITECTURES)

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

.KEEP_STATE:

all install lint clean: $(SUBDIRS)

clobber: $(SUBDIRS)
        $(RM) make.out lint.out

$(SUBDIRS): FRC
        @cd $@; pwd; $(MAKE) $(TARGET)

#
# Cross-reference customization: include all boot-related source files.
#
UTSDIR =        ../../../uts
UTSCLOSED =     ../../../../closed/uts
STANDLIBDIR =   ../../../stand/lib
STANDSYSDIRS =  ../../../stand/sys
PROMDIRS =      ../../promif
NAMESDIRS =     ../lib/names
XRDIRS +=       $(STANDLIBDIR) $(STANDSYSDIRS) $(PROMDIRS) $(NAMESDIRS)

#
# Components beginning with B! are in the open and closed trees; those
```



```
# beginning with O! are just in the open tree.
#
XRINCCOMP = B!sun4u O!sfmmu O!sparc/v7 O!sparc/v9 B!sparc B!sun B!common
XRINC_TMP = $(XRINCCOMP:B!%=$(UTSDIR)/%)
XRINCDIRS = $(XRINC_TMP:O!%=$(UTSDIR)/%)
$(CLOSED_BUILD)XRINC_TMP = $(XRINCCOMP:B!%=$(UTSDIR)/% $(UTSCLOSED)/%)
$(CLOSED_BUILD)XRINCDIRS = $(XRINC_TMP:O!%=$(UTSDIR)/%)

cscope.out tags: FRC
        $(XREF) -x $@

FRC:

# EXPORT DELETE START
EXPORT_SRC:
        $(RM) sparc/common/wanboot.c+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < sparc/common/wanboot.c > sparc/common/wanboot.c+
        $(MV) sparc/common/wanboot.c+ sparc/common/wanboot.c
        $(CHMOD) 444 sparc/common/wanboot.c
        $(RM) sparc/common/wbcli.c+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < sparc/common/wbcli.c > sparc/common/wbcli.c+
        $(MV) sparc/common/wbcli.c+ sparc/common/wbcli.c
        $(CHMOD) 444 sparc/common/wbcli.c
        $(RM) sparc/common/ramdisk.c+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < sparc/common/ramdisk.c > sparc/common/ramdisk.c+
        $(MV) sparc/common/ramdisk.c+ sparc/common/ramdisk.c
        $(CHMOD) 444 sparc/common/ramdisk.c
        $(RM) sparcv9/Makefile.com+
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < sparcv9/sun4/Makefile > sparcv9/sun4/Makefile+
        $(MV) sparcv9/sun4/Makefile+ sparcv9/sun4/Makefile
        $(CHMOD) 444 sparcv9/sun4/Makefile
        $(RM) Makefile+
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(RM) Makefile
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile
# EXPORT DELETE END
```

```
**********************************************************
   10177 Thu Jul 11 01:29:41 2013
new/usr/src/psm/stand/boot/sparc/common/ramdisk.c
first pass
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  23  * Use is subject to license terms.
  24  */

  26 #include <sys/param.h>
  27 #include <sys/promif.h>
  28 #include <sys/salib.h>
  29 /* EXPORT DELETE START */
  29 #include <bootlog.h>
  31 /* EXPORT DELETE END */
  30 #include "ramdisk.h"

  32 #include <sys/param.h>
  33 #include <sys/fcntl.h>
  34 #include <sys/obpdefs.h>
  35 #include <sys/reboot.h>
  36 #include <sys/promif.h>
  37 #include <sys/stat.h>
  38 #include <sys/bootvfs.h>
  39 #include <sys/platnames.h>
  40 #include <sys/salib.h>
  41 #include <sys/elf.h>
  42 #include <sys/link.h>
  43 #include <sys/auxv.h>
  44 #include <sys/boot_policy.h>
  45 #include <sys/boot_redirect.h>
  46 #include <sys/bootconf.h>
  47 #include <sys/boot.h>
  48 #include "boot_plat.h"


  51 static char ramdisk_preamble_fth[] =

  53 ": find-abort ( name$ -- ) "
  54 "   .\" Can't find \" type abort "
  55 "; "

  57 ": get-package ( pkg$ -- ph ) "
  58 "   2dup  find-package 0=  if "
  59 "      find-abort "
```



```
  60 "   then                        ( pkg$ ph ) "
  61 "   nip nip                     ( ph ) "
  62 "; "

  64 "\" /openprom/client-services\" get-package  constant cif-ph "

  66 "instance defer cif-open     ( dev$ -- ihandle|0 ) "
  67 "instance defer cif-close    ( ihandle -- ) "

  69 ": find-cif-method ( adr,len -- acf ) "
  70 "   2dup  cif-ph find-method 0=  if    ( adr,len ) "
  71 "      find-abort "
  72 "   then                            ( adr,len acf ) "
  73 "   nip nip                         ( acf ) "
  74 "; "

  76 "\" open\"     find-cif-method to cif-open "
  77 "\" close\"    find-cif-method to cif-close "

  79 "0 value dev-ih "

  81 "d# 100 buffer: open-cstr "

  83 ": dev-open ( dev$ -- okay? ) "
  84 /* copy to C string for open  */
  85 "   0  over open-cstr +  c! "
  86 "   open-cstr swap  move "
  87 "   open-cstr  cif-open dup  if "
  88 "      dup to dev-ih "
  89 "   then "
  90 "; "

  92 ": dev-close ( -- ) "
  93 "   dev-ih cif-close "
  94 "   0 to dev-ih "
  95 "; "

  97 ": open-abort  ( file$ -- ) "
  98 "   .\" Can't open \"  type  abort "
  99 "; "
 100 ;

 102 static char ramdisk_fth[] =

 104 "\" /\" get-package  push-package "

 106 "new-device "
 107 "   \" %s\" device-name "
 108 "    "
 109 "   \" block\"          device-type "
 110 "   \" SUNW,ramdisk\"   encode-string \" compatible\"  property"

 112 "   0 instance value current-offset "
 113 "    "
 114 "   0 value ramdisk-base-va "
 115 "   0 value ramdisk-size "
 116 "   0 value alloc-size "
 117 "    "
 118 "   : set-props "
 119 "      ramdisk-size     encode-int  \" size\"        property "
 120 "      ramdisk-base-va  encode-int  \" address\"     property "
 121 "      alloc-size       encode-int  \" alloc-size\"  property "
 122 "   ; "
 123 "   set-props "
 124 "    "
 125 "   : current-va  ( -- adr )  ramdisk-base-va current-offset +  ; "
```

```
126 "    "
127 "   external "
128 "    "
129 "   : open  ( -- okay? ) "
130 /* " .\" ramdisk-open\" cr " */
131 "      true "
132 "   ; "
133 "    "
134 "   : close  ( -- ) "
135 "   ; "
136 "    "
137 "   : seek  ( off.low off.high -- error? ) "
138 /* " 2dup .\" ramdisk-seek: \" .x .x " */
139 "      drop  dup  ramdisk-size  >  if "
140 /* " .\" fail\" cr " */
141 "         drop true  exit         ( failed ) "
142 "      then "
143 "      to current-offset  false   ( succeeded ) "
144 /* " .\" OK\" cr " */
145 "   ; "
146 "    "
147 "   : read  ( addr len -- actual-len ) "
148 /* " 2dup .\" ramdisk-read: \" .x .x " */
149 "      dup  current-offset  +           ( addr len new-off ) "
150 "      dup  ramdisk-size  >  if "
151 "         ramdisk-size -  -             ( addr len' ) "
152 "         ramdisk-size                  ( addr len new-off ) "
153 "      then  -rot                       ( new-off addr len ) "
154 "      tuck  current-va  -rot  move     ( new-off len ) "
155 "      swap  to current-offset          ( len ) "
156 /* " dup .x cr " */
157 "   ; "
158 "    "
159 "   : create ( alloc-sz base size -- ) "
160 "      to ramdisk-size "
161 "      to ramdisk-base-va "
162 "      to alloc-size "
163 "      set-props "
164 "   ; "
165 "    "
166 "finish-device "
167 "pop-package "

169 "\" /%s\" 2dup  dev-open  0=  if "
170 "   open-abort "
171 "then 2drop "

173 /* %x %x %x will be replaced by alloc-sz, base, size respectively */
174 "h# %x h# %x h# %x ( alloc-sz base size ) "
175 "\" create\" dev-ih  $call-method  (  ) "
176 "dev-close "

178 ;

180 char ramdisk_bootable[] =

182 "\" /chosen\" get-package  push-package "
183 "   \" nfs\"             encode-string  \" fstype\"  property "
184 "   \" /%s\"             encode-string  \" bootarchive\"  property "
185 "pop-package "

187 "   h# %x d# 512 +  to load-base init-program "
188 ;

190 #define BOOT_ARCHIVE_ALLOC_SIZE (32 * 1024 * 1024)      /* 32 MB */
191 #define BOOTFS_VIRT             ((caddr_t)0x50f00000)
```



```
192 #define ROOTFS_VIRT             ((caddr_t)0x52000000)

194 struct ramdisk_attr {
195         char *rd_name;
196         caddr_t rd_base;
197         size_t rd_size;
198 } ramdisk_attr[] = {
```

_____***unchanged_portion_omitted_***

****
45490 Thu Jul 11 01:29:41 2013
new/usr/src/psm/stand/boot/sparc/common/wanboot.c
first pass
****

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

#include <sys/types.h>
/* EXPORT DELETE START */
#include <sys/promif.h>
#include <sys/obpdefs.h>
#include <sys/bootvfs.h>
#include <sys/bootconf.h>
#include <netinet/in.h>
#include <sys/wanboot_impl.h>
#include <boot_http.h>
#include <aes.h>
#include <des3.h>
#include <cbc.h>
#include <hmac_sha1.h>
#include <sys/sha1.h>
#include <sys/sha1_consts.h>
#include <bootlog.h>
#include <parseURL.h>
#include <netboot_paths.h>
#include <netinet/inetutil.h>
#include <sys/salib.h>
#include <inet/mac.h>
#include <inet/ipv4.h>
#include <dhcp_impl.h>
#include <inet/dhcpv4.h>
#include <bootinfo.h>
#include <wanboot_conf.h>
#include "boot_plat.h"
#include "ramdisk.h"
#include "wbcli.h"

/*
 * Types of downloads
 */
#define MINIINFO        "miniinfo"
#define MINIROOT        "miniroot"
#define WANBOOTFS       "wanbootfs"
```



```
#define WANBOOT_RETRY_NOMAX     -1
#define WANBOOT_RETRY_ROOT_MAX  50
#define WANBOOT_RETRY_MAX       5
#define WANBOOT_RETRY_SECS      5
#define WANBOOT_RETRY_MAX_SECS  30

/*
 * Our read requests should timeout after 25 seconds
 */
#define SOCKET_READ_TIMEOUT     25

/*
 * Experimentation has shown that an 8K download buffer is optimal
 */
#define HTTP_XFER_SIZE          8192
static char     buffer[HTTP_XFER_SIZE];

bc_handle_t     bc_handle;

extern int      determine_fstype_and_mountroot(char *);
extern uint64_t get_ticks(void);

/*
 * The following is used to determine whether the certs and private key
 * files will be in PEM format or PKCS12 format.  'use_p12' is zero
 * to use PEM format, and 1 when PKCS12 format is to be used.  It is
 * done this way, as a global, so that it can be patched if needs be
 * using the OBP debugger.
 */
uint32_t        use_p12 = 1;

#define CONTENT_LENGTH          "Content-Length"

#define NONCELEN        (2 * HMAC_DIGEST_LEN) /* two hex nibbles/byte */
#define WANBOOTFS_NONCE_FILE    "/nonce"

static char nonce[NONCELEN + 1];

enum URLtype {
        URLtype_wanbootfs = 0,
        URLtype_miniroot = 1
};
______unchanged_portion_omitted_

/*
 * This implementation of bootprog() is used solely by wanboot.
 *
 * The basic algorithm is as follows:
 *
 * - The wanboot options (those specified using the "-o" flag) are processed,
 *   and if necessary the wanboot interpreter is invoked to collect other
 *   options.
 *
 * - The wanboot filesystem (containing certificates, wanboot.conf file, etc.)
 *   is then downloaded into the bootfs ramdisk, which is mounted for use
 *   by OpenSSL, access to wanboot.conf, etc.
 *
 * - The wanboot miniroot is downloaded over http/https into the rootfs
 *   ramdisk.  The bootfs filesystem is unmounted, and the rootfs filesystem
 *   is booted.
 */
/* EXPORT DELETE END */
/*ARGSUSED*/
int
bootprog(char *bpath, char *bargs, boolean_t user_specified_filename)
```

```
1592 {
1595 /* EXPORT DELETE START */
1593         char            *miniroot_path;
1594         url_t           server_url;
1595         int             ret;

1597         if (!init_netdev(bpath)) {
1598                 return (-1);
1599         }

1601         if (!bootinfo_init()) {
1602                 bootlog("wanboot", BOOTLOG_CRIT, "Cannot initialize bootinfo");
1603                 return (-1);
1604         }

1606         /*
1607          * Get default values from PROM, etc., process any boot arguments
1608          * (specified with the "-o" option), and initialize the interface.
1609          */
1610         if (!wanboot_init_interface(wanboot_arguments)) {
1611                 return (-1);
1612         }

1614         /*
1615          * Determine which encryption and hashing algorithms the client
1616          * is configured to use.
1617          */
1618         init_encryption();
1619         init_hashing();

1621         /*
1622          * Get the bootserver value.  Should be of the form:
1623          *      http://host[:port]/abspath.
1624          */
1625         ret = get_url(BI_BOOTSERVER, &server_url);
1626         if (ret != 0) {
1627                 bootlog("wanboot", BOOTLOG_CRIT,
1628                     "Unable to retrieve the bootserver URL");
1629                 return (-1);
1630         }

1632         /*
1633          * Get the wanboot file system and mount it. Contains metdata
1634          * needed by wanboot.
1635          */
1636         if (get_wanbootfs(&server_url) != 0) {
1637                 return (-1);
1638         }

1640         /*
1641          * Check that there is a valid wanboot.conf file in the wanboot
1642          * file system.
1643          */
1644         if (bootconf_init(&bc_handle, NULL) != BC_E_NOERROR) {
1645                 bootlog("wanboot", BOOTLOG_CRIT,
1646                     "wanboot.conf error (code=%d)", bc_handle.bc_error_code);
1647                 return (-1);
1648         }

1650         /*
1651          * Set the time
1652          */
1653         init_boot_time();

1655         /*
1656          * Verify that URLs in wanboot.conf can be reached, etc.
```



```
1657          */
1658         if (!wanboot_verify_config()) {
1659                 return (-1);
1660         }

1662         /*
1663          * Retrieve the miniroot.
1664          */
1665         if (get_miniroot(&miniroot_path) != 0) {
1666                 return (-1);
1667         }

1669         /*
1670          * We don't need the wanboot file system mounted anymore and
1671          * should unmount it so that we can mount the miniroot.
1672          */
1673         (void) unmountroot();

1675         boot_ramdisk(RD_ROOTFS);

1680 /* EXPORT DELETE END */
1677         return (0);
1678 }
```

_____unchanged_portion_omitted_

```
**********************************************************
   35075 Thu Jul 11 01:29:42 2013
new/usr/src/psm/stand/boot/sparc/common/wbcli.c
first pass
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
  23  * Use is subject to license terms.
  24  */

  26 #pragma ident   "%Z%%M% %I%     %E% SMI"

  28 /* EXPORT DELETE START */
  28 #include <sys/types.h>
  29 #include <sys/param.h>
  30 #include <sys/salib.h>
  31 #include <sys/promif.h>
  32 #include <sys/wanboot_impl.h>
  33 #include <netinet/in.h>
  34 #include <parseURL.h>
  35 #include <bootlog.h>
  36 #include <sys/socket.h>
  37 #include <netinet/inetutil.h>
  38 #include <netinet/dhcp.h>
  39 #include <dhcp_impl.h>
  40 #include <lib/inet/mac.h>
  41 #include <lib/inet/ipv4.h>
  42 #include <lib/inet/dhcpv4.h>
  43 #include <lib/sock/sock_test.h>
  44 #include <sys/sunos_dhcp_class.h>
  45 #include <aes.h>
  46 #include <des3.h>
  47 #include <hmac_sha1.h>
  48 #include <netdb.h>
  49 #include <wanboot_conf.h>
  50 #include <bootinfo.h>
  52 /* EXPORT DELETE END */

  52 #include "wbcli.h"

  56 /* EXPORT DELETE START */

  54 #define skipspace(p)    while (isspace(*(p))) ++p

  56 #define skiptext(p)     while (*(p) != '\0' && !isspace(*(p)) && \
  57                             *(p) != '=' && *(p) != ',') ++p
```



```
  59 #define PROMPT          "boot> "
  60 #define TEST_PROMPT     "boot-test> "

  62 #define CLI_SET         0
  63 #define CLI_FAIL        (-1)
  64 #define CLI_EXIT        (-2)
  65 #define CLI_CONT        (-3)

  67 #define CLF_CMD         0x00000001      /* builtin command */
  68 #define CLF_ARG         0x00000002      /* boot argument directive */

  70 #define CLF_IF          0x00000100      /* interface parameter */
  71 #define CLF_BM          0x00000200      /* bootmisc parameter */

  73 #define CLF_VALSET      0x00010000      /* value set, may be null */
  74 #define CLF_HIDDEN      0x00020000      /* don't show its value (key) */
  75 #define CLF_VALMOD      0x00040000      /* value modified by the user */

  77 /*
  78  * Macros for use in managing the flags in the cli_list[].
  79  * The conventions we follow are:
  80  *
  81  *      CLF_VALSET is cleared   if a value is removed from varptr
  82  *      CLF_VALSET is set       if a value has been placed in varptr
  83  *                              (that value need not be vetted)
  84  *      CLF_HIDDEN is set       if a value must not be exposed to the user
  85  *      CLF_HIDDEN is cleared   if a value can be exposed to the user
  86  *      CLF_VALMOD is cleared   if a value in varptr has not been modified
  87  *      CLF_VALMOD is set       if a value in varptr has been modified by
  88  *                              the user
  89  */
  90 #ifdef  DEBUG
  91 #define CLF_SETVAL(var)         {                                       \
  92                                         (((var)->flags) |= CLF_VALSET); \
  93                                         printf("set %s\n", var->varname);\
  94                                 }

  96 #define CLF_ISSET(var)          (printf("%s\n",                         \
  97                                     (((var)->flags) & CLF_VALSET) != 0  \
  98                                     ? "is set" : "not set"),            \
  99                                     ((((var)->flags) & CLF_VALSET) != 0))

 101 #define CLF_CLRHIDDEN(var)      {                                       \
 102                                         (((var)->flags) &= ~CLF_HIDDEN); \
 103                                         printf("unhide %s\n", var->varname); \
 104                                 }

 106 #define CLF_ISHIDDEN(var)       (printf("%s\n",                         \
 107                                     (((var)->flags) & CLF_HIDDEN) != 0  \
 108                                     ? "is hidden" : "not hidden"),      \
 109                                     ((((var)->flags) & CLF_HIDDEN) != 0))

 111 #define CLF_MODVAL(var)         {                                       \
 112                                         (((var)->flags) |=              \
 113                                         (CLF_VALMOD | CLF_VALSET));     \
 114                                         printf("modified %s\n", var->varname);\
 115                                 }

 117 #define CLF_ISMOD(var)          (printf("%s\n",                         \
 118                                     (((var)->flags) & CLF_VALMOD) != 0 \
 119                                     ? "is set" : "not set"),    \
 120                                     ((((var)->flags) & CLF_VALMOD) != 0))
 121 #else   /* DEBUG */

 123 #define CLF_SETVAL(var)         (((var)->flags) |= CLF_VALSET)
```

```
 124 #define CLF_ISSET(var)          ((((var)->flags) & CLF_VALSET) != 0)
 125 #define CLF_CLRHIDDEN(var)      (((var)->flags) &= ~CLF_HIDDEN)
 126 #define CLF_ISHIDDEN(var)       ((((var)->flags) & CLF_HIDDEN) != 0)
 127 #define CLF_MODVAL(var)         (((var)->flags) |= (CLF_VALMOD | CLF_VALSET))
 128 #define CLF_ISMOD(var)          ((((var)->flags) & CLF_VALMOD) != 0)

 130 #endif  /* DEBUG */

 132 /*
 133  * The width of the widest varname below - currently "subnet_mask".
 134  */
 135 #define VAR_MAXWIDTH    strlen(BI_SUBNET_MASK)

 137 struct cli_ent;
 138 typedef int claction_t(struct cli_ent *, char *, boolean_t);

 140 typedef struct cli_ent {
 141         char            *varname;
 142         claction_t      *action;
 143         int             flags;
 144         void            *varptr;
 145         uint_t          varlen;
 146         uint_t          varmax;
 147 } cli_ent_t;
_____unchanged_portion_omitted_

1308 /* EXPORT DELETE END */
1304 boolean_t
1305 wanboot_init_interface(char *boot_arguments)
1306 {
1312 /* EXPORT DELETE START */
1307         boolean_t       interactive;
1308         int             which;

1310 #if     defined(__sparcv9)
1311         /*
1312          * Get the keys from PROM before we allow the user
1313          * to override them from the CLI.
1314          */
1315         get_prom_encr_keys();
1316         get_prom_hash_keys();
1317 #endif  /* defined(__sparcv9) */

1319         /*
1320          * If there is already a bootp-response property under
1321          * /chosen then the PROM must have done DHCP for us;
1322          * invoke dhcp() to 'bind' the interface.
1323          */
1324         if (bootinfo_get(BI_BOOTP_RESPONSE, NULL, NULL, NULL) ==
1325             BI_E_BUF2SMALL) {
1326                 (void) cldhcp(NULL, NULL, 0);
1327         }

1329         /*
1330          * Obtain default interface values from bootinfo.
1331          */
1332         bootinfo_defaults(CLF_IF);

1334         /*
1335          * Process the boot arguments (following the "-o" option).
1336          */
1337         if (boot_arguments != NULL) {
1338                 (void) cli_eval_buf(boot_arguments,
1339                     (CLF_ARG | CLF_IF | CLF_BM));
1340         }
```



```
1342         /*
1343          * Stash away any interface/bootmisc parameter values we got
1344          * from either the PROM or the boot arguments.
1345          */
1346         update_bootinfo(CLF_IF | CLF_BM);

1348         /*
1349          * If we don't already have a value for bootserver, try to
1350          * deduce one.  Refresh wbcli's idea of these values.
1351          */
1352         determine_bootserver_url();
1353         bootinfo_defaults(CLF_BM);

1355         /*
1356          * Check that the information we have collected thus far is sufficient.
1357          */
1358         interactive = args_specified_prompt;

1360         if (interactive) {
1361                 /*
1362                  * Drop into the boot interpreter to allow the input
1363                  * of keys, bootserver and bootmisc, and in the case
1364                  * that net-config-strategy == "manual" the interface
1365                  * parameters.
1366                  */
1367                 which = CLF_BM | CLF_CMD;
1368                 if (strcmp(net_config_strategy(), "manual") == 0)
1369                         which |= CLF_IF;

1371                 do {
1372                         cli_interpret(which);
1373                         update_bootinfo(CLF_IF | CLF_BM);
1374                 } while (config_incomplete(CLF_IF, interactive));
1375         } else {
1376                 /*
1377                  * The user is not to be given the opportunity to
1378                  * enter further values; fail.
1379                  */
1380                 if (config_incomplete(CLF_IF, interactive)) {
1381                         bootlog("wanboot", BOOTLOG_CRIT,
1382                             "interface incorrectly configured");
1383                         return (B_FALSE);
1384                 }
1385         }

1387         /*
1388          * If a wanboot-enabled PROM hasn't processed client-id in
1389          * network-boot-arguments, or no value for client-id has been
1390          * specified to the boot interpreter, then provide a default
1391          * client-id based on our MAC address.
1392          */
1393         generate_default_clientid();

1395         /*
1396          * If net-config-strategy == "manual" then we must setup
1397          * the interface now; if "dhcp" then it will already have
1398          * been setup.
1399          */
1400         if (strcmp(net_config_strategy(), "manual") == 0)
1401                 setup_interface();
1408 /* EXPORT DELETE END */
1402         return (B_TRUE);
1403 }

1405 boolean_t
1406 wanboot_verify_config(void)
```

```
1407 {
1415 /* EXPORT DELETE START */
1408         /*
1409          * Check that the wanboot.conf file defines a valid root_server
1410          * URL, and check that, if given, the boot_logger URL is valid.
1411          */
1412         if (config_incomplete(0, B_FALSE)) {
1413                 bootlog("wanboot", BOOTLOG_CRIT,
1414                     "incomplete boot configuration");
1415                 return (B_FALSE);
1416         }
1425 /* EXPORT DELETE END */
1417         return (B_TRUE);
1418 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    5954 Thu Jul 11 01:29:43 2013
new/usr/src/psm/stand/boot/sparcv9/sun4/Makefile
first pass
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #

  26 TOPDIR =        ../../../../..

  28 include $(TOPDIR)/uts/Makefile.uts

  30 all     :=      TARGET = all
  31 install :=      TARGET = install
  32 clean   :=      TARGET = clean

  34 TARG_MACH =     sparcv9
  35 TARG_MACH_DIR = sparcv9
  36 ARCHVERS =      v9
  37 PLATFORM =      sun4
  38 #ARCHMMU  =     sfmmu
  39 PROMVERS =      ieee1275
  40 ASFLAGS +=      $(sparcv9_XARCH)

  42 ARCH_C_SRC =    sun4u_memlist.c sun4x_standalloc.c sun4dep.c
  43 ARCH_S_SRC =    sparcv9_subr.s
  44 SRT0_S =        sun4u_srt0.s
  45 INLINES =

  47 LDFLAGS +=      -L$(TOPDIR)/psm/stand/lib/promif/$(TARG_MACH)/$(PROMVERS)/common

  49 #
  50 # The following libraries are build in LIBPLAT_DIR
  51 #
  52 LIBPLAT_DIR =   $(TOPDIR)/psm/stand/lib/promif/$(TARG_MACH)/$(PROMVERS)/$(PLATFO
  53 LIBPLAT_LIBS =  libplat.a
  54 LIBPLAT_L_LIBS= $(LIBPLAT_LIBS:lib%.a=llib-l%.ln)
  55 LIBPLAT_DEP =   $(LIBPLAT_DIR)/$(LIBPLAT_LIBS)
  56 LIBPLAT_DEP_L = $(LIBPLAT_DIR)/$(LIBPLAT_L_LIBS)

  58 #
  59 # Platform specific libraries
  60 #
  61 PSMLIBS +=      $(LIBPLAT_LIBS:lib%.a=-l%)
```



```
  62 PSMLIB_DIRS +=  $(LIBPLAT_DIR)

  64 include ../Makefile.com

  66 CPPINCS         += -I$(TOPDIR)/psm/stand/boot/sparc/sun4

  68 #
  69 # Set the choice of compiler.

  71 include $(TOPDIR)/psm/Makefile.psm.64

  73 CFLAGS64        += -xchip=ultra $(CCABS32)

  75 #
  76 # XXX this totally sucks since it effectively turns off -errchk=longptr64,
  77 # which we really should be using.
  78 #
  79 LINTFLAGS64 = $(LINTFLAGS) -m64

  81 #
  82 # Cross-reference customization: include all boot-related source files.
  83 #
  84 STANDLIBDIR=    ../../../../../stand/lib
  85 STANDSYSDIR=    ../../../../../stand/sys
  86 PROMDIRS=       ../../../../promif
  87 NAMESDIRS=      ../../../lib/names/sparcv9 ../../../lib/names/sparc/common
  88 XRDIRS +=       ../../sparc/common ../../common $(STANDLIBDIR) \
  89                 $(STANDSYSDIR) $(PROMDIRS) $(NAMESDIRS)
  90 XRPRUNE =       i86pc i386


  93 ##############################
  94 #
  95 #  WANboot booter
  96 #
  97 # Libraries used to build wanboot
  98 #
  99 # EXPORT DELETE START
 100 LIBWANBOOT =    libwanboot.a
 101 LIBSCRYPT =     libscrypt.a
 102 LIBSSL =        libssl.a
 103 LIBCRYPTO =     libcrypto.a
 104 # EXPORT DELETE END

  99 LIBWAN_LIBS     = \
 100                 libwanboot.a \
 107                 $(LIBWANBOOT) \
 101                 libnvpair.a libufs.a libhsfs.a libnfs.a \
 102                 libxdr.a libnames.a libsock.a libinet.a libtcp.a \
 103                 libscrypt.a libssl.a libcrypto.a \
 110                 $(LIBSCRYPT) $(LIBSSL) $(LIBCRYPTO) \
 104                 libmd5.a libsa.a libprom.a \
 112                 $(LIBSSL) \
 105                 $(LIBPLAT_LIBS)
 106 WAN_LIBS        = $(LIBWAN_LIBS:lib%.a=-l%)
 107 WAN_DIRS        = $(LIBNAME_DIR:%=-L%) $(LIBSYS_DIR:%=-L%)
 108 WAN_DIRS        += $(LIBPLAT_DIR:%=-L%) $(LIBPROM_DIR:%=-L%)

 110 #
 111 # Loader flags used to build wanboot
 112 #
 113 WAN_MAPFILE     = $(MACH_DIR)/mapfile
 114 WAN_LDFLAGS     = -dn -M $(WAN_MAPFILE) -e _start $(WAN_DIRS)
 115 WAN_L_LDFLAGS   = $(WAN_DIRS)

 117 #
```

```
118 # Object files used to build wanboot
119 #
120 WAN_SRT0        = $(SRT0_OBJ)
121 WAN_OBJS        = $(OBJS) wbfsconf.o wbcli.o wanboot.o ramdisk.o
122 WAN_L_OBJS      = $(WAN_SRT0:%.o=%.ln) $(WAN_OBJS:%.o=%.ln)


125 ##############################
126 #
127 # NFS booter
128 #
129 # Libraries used to build nfsboot
130 #
131 LIBNFS_LIBS     = libnfs.a libxdr.a libnames.a \
132                 libsock.a libinet.a libtcp.a libsa.a libprom.a \
133                 $(LIBPLAT_LIBS)
134 NFS_LIBS        = $(LIBNFS_LIBS:lib%.a=-l%)
135 NFS_DIRS        = $(LIBNAME_DIR:%=-L%) $(LIBSYS_DIR:%=-L%)
136 NFS_DIRS        += $(LIBPLAT_DIR:%=-L%) $(LIBPROM_DIR:%=-L%)

138 #
139 # Loader flags used to build inetboot
140 #
141 NFS_MAPFILE     = $(MACH_DIR)/mapfile
142 NFS_LDFLAGS     = -dn -M $(NFS_MAPFILE) -e _start $(NFS_DIRS)
143 NFS_L_LDFLAGS   = $(NFS_DIRS)

145 #
146 # Object files used to build inetboot
147 #
148 NFS_SRT0        = $(SRT0_OBJ)
149 NFS_OBJS        = $(OBJS) nfsconf.o inetboot.o ramdisk.o
150 NFS_L_OBJS      = $(NFS_SRT0:%.o=%.ln) $(NFS_OBJS:%.o=%.ln)


153 #include $(BOOTSRCDIR)/Makefile.rules

155 FRC:

157 .KEEP_STATE:

159 all: $(WANBOOT) $(NFSBOOT)

161 install: all                                    \
162         $(ROOT_PLAT_SUN4U_WANBOOT)              \
163         $(ROOT_PLAT_SUN4V_WANBOOT)              \
164         $(USR_PLAT_SUN4U_LIB_FS_NFS_NFSBOOT)    \
165         $(USR_PLAT_SUN4V_LIB_FS_NFS_NFSBOOT)

167 $(WANBOOT): $(WAN_MAPFILE) $(WAN_SRT0) $(WAN_OBJS) $(LIBDEPS)
168         $(LD) $(WAN_LDFLAGS) -o $@ $(WAN_SRT0) $(WAN_OBJS) $(WAN_LIBS)
169         $(MCS) -d $@
170         $(POST_PROCESS)
171         $(MCS) -c $@
172         $(STRIP) $@

174 $(NFSBOOT): $(NFS_MAPFILE) $(NFS_SRT0) $(NFS_OBJS) $(LIBDEPS)
175         $(LD) $(NFS_LDFLAGS) -o $@ $(NFS_SRT0) $(NFS_OBJS) $(NFS_LIBS)
176         $(MCS) -d $@
177         $(POST_PROCESS)
178         $(MCS) -c $@
179         $(STRIP) $@

181 $(WANBOOT)_lint: $(WAN_L_OBJS) $(L_LIBDEPS)
182         @echo ""
183         @echo wanboot lint: global crosschecks:
```



```
184         $(LINT.c) $(WAN_L_LDFLAGS) $(WAN_L_OBJS) $(WAN_LIBS)

186 $(NFSBOOT)_lint: $(NFS_L_OBJS) $(L_LIBDEPS)
187         @echo ""
188         @echo inetboot lint: global crosschecks:
189         $(LINT.c) $(NFS_L_LDFLAGS) $(NFS_L_OBJS) $(NFS_LIBS)

191 $(ROOT_PLAT_SUN4U_WANBOOT): $(WANBOOT)
192         $(INS) -s -m $(FILEMODE) -f $(ROOT_PLAT_DIR)/sun4u $(WANBOOT)

194 $(ROOT_PLAT_SUN4V_WANBOOT): $(WANBOOT)
195         $(INS) -s -m $(FILEMODE) -f $(ROOT_PLAT_DIR)/sun4v $(WANBOOT)

197 $(USR_PLAT_SUN4U_LIB_FS_NFS):
198         $(INS.dir)

200 $(USR_PLAT_SUN4V_LIB_FS_NFS):
201         $(INS.dir)

203 $(USR_PLAT_SUN4U_LIB_FS_NFS_NFSBOOT): $(USR_PLAT_SUN4U_LIB_FS_NFS) $(NFSBOOT)
204         $(INS) -s -m $(FILEMODE) -f $(USR_PLAT_SUN4U_LIB_FS_NFS) $(NFSBOOT)

206 $(USR_PLAT_SUN4V_LIB_FS_NFS_NFSBOOT): $(USR_PLAT_SUN4V_LIB_FS_NFS) $(NFSBOOT)
207         $(INS) -s -m $(FILEMODE) -f $(USR_PLAT_SUN4V_LIB_FS_NFS) $(NFSBOOT)

209 $(STRIPALIGN): $(CMN_DIR)/$$(@).c
210         $(NATIVECC) -o $@ $(CMN_DIR)/$@.c

212 clean:
213         $(RM) make.out lint.out
214         $(RM) $(OBJS) $(CONF_OBJS) $(MISC_OBJS) $(SRT0_OBJ)
215         $(RM) $(WANBOOT_OBJS) $(NFSBOOT_OBJS)
216         $(RM) $(L_OBJS) $(CONF_L_OBJS) $(MISC_L_OBJS) $(SRT0_L_OBJ)
217         $(RM) $(WANBOOT_L_OBJS) $(NFSBOOT_L_OBJS)

219 clobber: clean
220         $(RM) $(WANBOOT) $(NFSBOOT) $(STRIPALIGN)

222 lint: $(WANBOOT)_lint $(NFSBOOT)_lint
```

```
**********************************************************
    1404 Thu Jul 11 01:29:43 2013
new/usr/src/req.flg
first pass
**********************************************************
```

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

echo_file usr/src/Makefile
echo_file usr/src/Targetdirs
echo_file usr/src/Makefile.master
echo_file usr/src/Makefile.noget
echo_file usr/src/Makefile.master.64
echo_file usr/src/Makefile.msg.targ
echo_file usr/src/Makefile.psm
echo_file usr/src/Makefile.psm.targ
echo_file usr/src/xmod/xmod_files
echo_file usr/closed/cmd/cmd-crypto/etc/certs/SUNWosnetCF
echo_file usr/closed/cmd/cmd-crypto/etc/certs/SUNWosnetSE
echo_file usr/closed/cmd/cmd-crypto/etc/keys/SUNWosnetCF
echo_file usr/closed/cmd/cmd-crypto/etc/keys/SUNWosnetSE
```

```
**********************************************************
    8212 Thu Jul 11 01:29:44 2013
new/usr/src/tools/findunref/exception_list.open
first pass
**********************************************************
```

```
#
# 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) 2001, 2010, Oracle and/or its affiliates. All rights reserved.
#

#
# open-tree exception list
#
# See README.exception_lists for details
#

#
# Ignore oddly-named text files scattered about -- someday these should be
# suffixed with .txt so we don't have to list them.
#
./usr/src/cmd/oawk/EXPLAIN
./usr/src/cmd/vi/port/ex.news
./usr/src/cmd/ssh/doc

#
# Ignore everything under trees that may be resynched from outside ON.
#
./usr/src/cmd/perl
./usr/src/cmd/sqlite
./usr/src/lib/libsqlite
./usr/src/cmd/tcpd
./usr/src/common/openssl
./usr/src/grub
./usr/src/uts/intel/sys/acpi

#
# Ignore ksh93/ast-related files that are only used to resync our build
# configuration with upstream.
#
./usr/src/lib/libast/*/src/lib/libast/FEATURE
./usr/src/lib/libast/*/src/lib/libast/ast_namval.h
./usr/src/lib/libast/common/comp/conf.*
./usr/src/lib/libast/common/features
./usr/src/lib/libast/common/include/ast_windows.h
./usr/src/lib/libast/common/port/lc.tab
./usr/src/lib/libast/common/port/lcgen.c
```



```
./usr/src/lib/libcmd/*/src/lib/libcmd/FEATURE
./usr/src/lib/libcmd/common/features
./usr/src/lib/libdll/*/src/lib/libdll/FEATURE
./usr/src/lib/libdll/common/features
./usr/src/lib/libpp/*/pp.*
./usr/src/lib/libpp/common/gentab.sh
./usr/src/lib/libpp/common/ppsym.c
./usr/src/lib/libpp/i386/ppdebug.h
./usr/src/lib/libpp/sparc/ppdebug.h
./usr/src/lib/libshell/*/src/cmd/ksh93/FEATURE
./usr/src/lib/libshell/common/data/math.tab
./usr/src/lib/libshell/common/features
./usr/src/lib/libshell/misc/buildksh93.sh
./usr/src/lib/libshell/misc/buildksh93.readme

#
# Ignore ksh93/ast-related "iffe" (if feature enabled) probe
#
./usr/src/lib/libsum/common/features/sum

#
# Ignore ksh93/ast-related upstream source, currently superseded by
# a per-platform version of sum.h, since we use libmd.so.1 for some
# ciphers.
#
./usr/src/lib/libsum/common/sum.h

#
# Ignore ksh93/ast-related test programs.
#
./usr/src/cmd/ast/msgcc/msgcc.tst
./usr/src/lib/libast/common/port/astmath.c

#
# Ignore ksh93/ast-related source components that are not currently
# used but may be useful later.
#
./usr/src/lib/libcmd/common/cksum.c
./usr/src/lib/libcmd/common/md5sum.c
./usr/src/lib/libcmd/common/sum.c
./usr/src/lib/libshell/common/bltins/mkservice.c
./usr/src/lib/libshell/common/data/bash_pre_rc.sh
./usr/src/lib/libshell/common/include/env.h
./usr/src/lib/libshell/common/sh/bash.c
./usr/src/lib/libshell/common/sh/env.c
./usr/src/lib/libshell/common/sh/shcomp.c
./usr/src/lib/libshell/common/sh/suid_exec.c

#
# Ignore any files built as part of the nightly program itself.
#
# ISUSED - let checkpaths know that the next entry is good.
./usr/src/*.out
# ISUSED - let checkpaths know that the next entry is good.
./usr/src/*.ref

#
# Ignore internal test directories and test programs.
#
*/tests
*/test
*Test.java
*_test.[ch]
./usr/src/cmd/ldap/common/*test.c
./usr/src/cmd/logadm/tester
./usr/src/cmd/print/printmgr/com/sun/admin/pm/client/helptools/extract
```

```
./usr/src/cmd/print/printmgr/com/sun/admin/pm/server/pmtest
./usr/src/cmd/sendmail/libsm/t-*.c
./usr/src/cmd/sort/common/convert.c
./usr/src/cmd/sort/common/invoke.c
./usr/src/lib/crypt_modules/sha256/test.c
./usr/src/lib/efcode/fcode_test
./usr/src/lib/libkvm/common/test.c
./usr/src/lib/libsaveargs/tests/

#
# Ignore debugging code.
#
./usr/src/cmd/fs.d/pcfs/fsck/inject.c
./usr/src/cmd/sort/common/statistics.c

#
# Ignore internal packages, scripts, and tools that are intentionally not
# built or used during a nightly.
#
./usr/src/cmd/sgs/packages
./usr/src/cmd/sgs/rtld.4.x
./usr/src/prototypes
./usr/src/cmd/pools/poold/com/sun/solaris/*/*/package.html
./usr/src/uts/intel/io/acpica/cmp_ca.sh

#
# Ignore files that are only used by internal packages.
#
./usr/src/cmd/sgs/*/*/*chk.msg

#
# Ignore files that get used during a EXPORT_SRC or CRYPT_SRC build only.
#
./usr/src/common/crypto/aes/Makefile
./usr/src/common/crypto/arcfour/Makefile
./usr/src/common/crypto/blowfish/Makefile
./usr/src/common/crypto/des/Makefile
./usr/src/common/crypto/rsa/Makefile
./usr/src/lib/gss_mechs/mech_dh/backend/mapfile-vers
./usr/src/lib/gss_mechs/mech_dh/dh1024/mapfile-vers
./usr/src/lib/gss_mechs/mech_dh/dh192/mapfile-vers
./usr/src/lib/gss_mechs/mech_dh/dh640/mapfile-vers
./usr/src/lib/gss_mechs/mech_krb5/mapfile-vers-clean
./usr/src/lib/gss_mechs/mech_spnego/mapfile-vers-clean
./usr/src/lib/pkcs11/pkcs11_softtoken/common/Makefile
./usr/src/uts/common/Makefile
./usr/src/uts/common/crypto/io/Makefile
./usr/src/uts/common/gssapi/include/Makefile
./usr/src/uts/common/gssapi/mechs/dummy/Makefile
./usr/src/uts/common/gssapi/mechs/krb5/Makefile
./usr/src/xmod

#
# Ignore Makefiles which are used by developers but not used by nightly
# itself.  This is a questionable practice, since they tend to rot.
#
./usr/src/cmd/syslogd/sparcv9/Makefile
./usr/src/uts/sparc/uhci/Makefile
./usr/src/lib/pam_modules/smb/amd64/Makefile
./usr/src/lib/pam_modules/smb/sparcv9/Makefile
./usr/src/cmd/isns/isnsd/xml_def/isnsmgmtSchema.xsd

#
# Ignore dtrace scripts only used by developers
#
./usr/src/cmd/vscan/vscand/vscan.d
```



```
#
# Ignore sample source code.
#
./usr/src/cmd/sendmail/libmilter/example.c
./usr/src/lib/libdhcpsvc/modules/templates

#
# Ignore .xcl files that aren't used because the program is statically linked.
#
./usr/src/cmd/cmd-inet/sbin/dhcpagent/dhcpagent.xcl

#
# Ignore sendmail files included for completeness' sake, but which won't
# be used until certain _FFR (for future release) #define's go live.
#
./usr/src/cmd/sendmail/src/statusd_shm.h

#
# Ignore files originally supplied by ISC (Internet Software Consortium)
# as part of a BIND release.
#
./usr/src/lib/libresolv2/common/irs/getaddrinfo.c
./usr/src/lib/libresolv2/common/irs/nis_p.h
./usr/src/lib/libresolv2/common/resolv/res_mkupdate.h
./usr/src/lib/libresolv2/include/err.h

#
# Ignore mont_mulf.c. It is used as a starting point for some hand optimized
# assembly files. We keep it around for future reference.
#
./usr/src/common/bignum/mont_mulf.c

#
# Ignore the sparc Makefiles for x86-only drivers;
# they're used to build warlock only.
#
./usr/src/uts/sparc/sata/Makefile
./usr/src/uts/sparc/si3124/Makefile
./usr/src/uts/sparc/nv_sata/Makefile
./usr/src/uts/sparc/ahci/Makefile

#
# Ignore uttrack.c.  It is provided as part of the standard
# ACPI CA source code but provides optional resource tracking
# functionality which is not used.
#
./usr/src/uts/intel/io/acpica/utilities/uttrack.c

#
# Ignore any files that get used during a gcc build only.
#
./usr/src/cmd/sgs/rtld/common/mapfile-order-gcc

#
# The sharemgr command is built 32-bit only by default, but support
# for building 64-bit is latent in the Makefiles.
#
./usr/src/cmd/dfs.cmds/sharemgr/amd64/Makefile
./usr/src/cmd/dfs.cmds/sharemgr/sparcv9/Makefile

#
# Legitimately unreferenced license/copying files.  Please include
# explanatory comments when adding items here.
#
```

```
259 #
260 # OPENSOLARIS.LICENSE needs to remain in usr/src as long as it is
261 # referenced in the CDDL headers.
262 #
263 ./usr/src/OPENSOLARIS.LICENSE

265 # Though "COPYING" is usually used as a filename for GPL, the license
266 # information for openssh is actually found in usr/src/cmd/ssh/doc/LICENSE.
267 # The COPYING.Ylonen file is merely additional information.
268 #
269 ./usr/src/cmd/ssh/doc/COPYING.Ylonen

271 #
272 # This covers header files that are not delivered.
273 #
274 ./usr/src/uts/common/xen/public/COPYING

276 #
277 # ld tests which are not currently delivered
278 #
279 ./usr/src/cmd/sgs/test
```

```
**********************************************************
    4186 Thu Jul 11 01:29:45 2013
new/usr/src/tools/scripts/checkpaths.sh
first pass
**********************************************************
```

```
  1 #!/bin/ksh -p
  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 2009 Sun Microsystems, Inc.  All rights reserved.
 25 # Use is subject to license terms.
 26 #

 28 # Quis custodiet ipsos custodies?

 30 if [ -z "$SRC" ]; then
 31         SRC=$CODEMGR_WS/usr/src
 32 fi

 34 if [ -z "$CODEMGR_WS" -o ! -d "$CODEMGR_WS" -o ! -d "$SRC" ]; then
 35         echo "$0: must be run from within a workspace."
 36         exit 1
 37 fi

 39 cd $CODEMGR_WS || exit 1

 41 # Use -b to tell this script to ignore derived (built) objects.
 42 if [ "$1" = "-b" ]; then
 43         b_flg=y
 44 fi

 46 # Not currently used; available for temporary workarounds.
 47 args="-k NEVER_CHECK"

 49 # We intentionally don't depend on $MACH here, and thus no $ROOT.  If
 50 # a proto area exists, then we use it.  This allows this script to be
 51 # run against gates (which should contain both SPARC and x86 proto
 52 # areas), build workspaces (which should contain just one proto area),
 53 # and unbuilt workspaces (which contain no proto areas).
 54 if [ "$b_flg" = y ]; then
 55         rootlist=
 56 elif [ $# -gt 0 ]; then
 57         rootlist=$*
 58 else
 59         rootlist="$CODEMGR_WS/proto/root_sparc $CODEMGR_WS/proto/root_i386"
 60 fi
```



```
 62 # If the closed source is not present, then exclude IKE from validation.
 63 if [ "$CLOSED_IS_PRESENT" = no ]; then
 64         excl="-e ^usr/include/ike/"
 65 fi

 67 for ROOT in $rootlist
 68 do
 69         case "$ROOT" in
 70         *sparc|*sparc-nd)
 71                 arch=sparc
 72                 ;;
 73         *i386|*i386-nd)
 74                 arch=i386
 75                 ;;
 76         *)
 77                 echo "$ROOT has unknown architecture." >&2
 78                 exit 1
 79                 ;;
 80         esac
 81         if [ -d $ROOT ]; then
 82                 #
 83                 # This is the old-style packaging exception list, from
 84                 # the svr4-specific usr/src/pkgdefs
 85                 #
 86                 [ -f $SRC/pkgdefs/etc/exception_list_$arch ] && \
 87                         validate_paths '-s/\s*'$arch'$//' $excl -b $ROOT \
 88                             $args $SRC/pkgdefs/etc/exception_list_$arch
 89                 #
 90                 # These are the new-style packaging exception lists,
 91                 # from the repository-wide exception_lists/ directory.
 92                 #
 93                 e="$CODEMGR_WS/exception_lists/packaging"
 94                 for f in $e; do
 95                         if [ -f $f ]; then
 96                                 nawk 'NF == 1 || /[     ]\+'$arch'$/ { print; }'
 97                                     < $f | validate_paths -b $ROOT -n $f
 98                         fi
 99                 done
100         fi
101 done

103 # Two entries in the findunref exception_list deal with things created
104 # by nightly.  Otherwise, this test could be run on an unmodifed (and
105 # unbuilt) workspace.  We handle this by flagging the one that is
106 # present only on a built workspace (./*.out) and the one that's
107 # present only after a run of findunref (./*.ref) with ISUSED, and
108 # disabling all checks of them.  The assumption is that the entries
109 # marked with ISUSED are always known to be good, thus the Latin quote
110 # at the top of the file.
111 #
112 # The exception_list is generated from whichever input files are appropriate
113 # for this workspace, so checking it obviates the need to check the inputs.

115 if [ -r $SRC/tools/findunref/exception_list ]; then
116         validate_paths -k ISUSED -r -e '^\*' $SRC/tools/findunref/exception_list
117 fi

119 # These are straightforward.
120 if [ -d $SRC/xmod ]; then
121         # If the closed source is not present, then don't validate it.
122         if [ "$CLOSED_IS_PRESENT" = no ]; then
123                 excl_cry="-e ^usr/closed"
124                 excl_xmod="-e ^../closed"
125         fi
126         validate_paths $excl_cry $SRC/xmod/cry_files
127         validate_paths $excl_xmod -b $SRC $SRC/xmod/xmod_files
```

```
128 fi

119 if [ -f $SRC/tools/opensolaris/license-list ]; then
120         excl=
121         if [ "$CLOSED_IS_PRESENT" = no ]; then
122                 excl="-e ^usr/closed"
123         fi
124         sed -e 's/$/.descrip/' < $SRC/tools/opensolaris/license-list | \
125                 validate_paths -n SRC/tools/opensolaris/license-list $excl
126 fi

128 # Finally, make sure the that (req|inc).flg files are in good shape.
129 # If SCCS files are not expected to be present, though, then don't
130 # check them.
131 if [ ! -d "$CODEMGR_WS/Codemgr_wsdata" ]; then
132         f_flg='-f'
133 fi
134 # If the closed source is not present, then don't validate it.
135 if [ "$CLOSED_IS_PRESENT" = no ]; then
136         excl="-e ^usr/closed/"
137 fi

139 validate_flg $f_flg $excl

141 exit 0
```

```
**********************************************************
   18983 Thu Jul 11 01:29:45 2013
new/usr/src/tools/scripts/nightly.1
first pass
**********************************************************
```

```
   1 .\" "
   2 .\" " The contents of this file are subject to the terms of the
   3 .\" " Common Development and Distribution License (the "License").
   4 .\" " You may not use this file except in compliance with the License.
   5 .\" "
   6 .\" " You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   7 .\" " or http://www.opensolaris.org/os/licensing.
   8 .\" " See the License for the specific language governing permissions
   9 .\" " and limitations under the License.
  10 .\" "
  11 .\" " When distributing Covered Code, include this CDDL HEADER in each
  12 .\" " file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  13 .\" " If applicable, add the following below this CDDL HEADER, with the
  14 .\" " fields enclosed by brackets "[]" replaced with your own identifying
  15 .\" " information: Portions Copyright [yyyy] [name of copyright owner]
  16 .\" "
  17 .\" " CDDL HEADER END
  18 .\" "
  19 .\" "Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved
  20 .\" "Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
  21 .\" "
  22 .TH nightly 1 "6 July 2010"
  23 .SH NAME
  24 .I nightly
  25 \- build an OS-Net consolidation overnight
  26 .SH SYNOPSIS
  27 \fBnightly [-in] [-V VERS] <env_file>\fP
  28 .LP
  29 .SH DESCRIPTION
  30 .IX "OS-Net build tools" "nightly" "" "\fBnightly\fP"
  31 .LP
  32 .I nightly,
  33 the mother of all build scripts,
  34 can bringover, build, archive, package, error check, and
  35 generally do everything it takes to
  36 turn OS/Net consolidation source code into useful stuff.
  37 It is customizable to permit you to run anything from a
  38 simple build to all of the cross-checking a gatekeeper
  39 needs.  The advantage to using
  40 .I nightly
  41 is that you build things correctly, consistently and
  42 automatically, with the best practices; building with
  43 .I nightly
  44 can mean never having to say you’re sorry to your
  45 gatekeeper.
  46 .LP
  47 More
  48 specifically,
  49 .I nightly
  50 performs the following tasks, in order, if
  51 all these things are desired:
  52 .LP
  53 .RS
  54 .TP
  55 \(bu
  56 perform a "make clobber" to clean up old binaries
  57 .TP
  58 \(bu
  59 bringover from the identified parent gate/clone
  60 .TP
  61 \(bu
```



```
  62 perform non-DEBUG and DEBUG builds
  63 .TP
  64 \(bu
  65 list proto area files and compare with previous list
  66 .TP
  67 \(bu
  68 copy updated proto area to parent
  69 .TP
  70 \(bu
  71 list shared lib interface and compare with previous list
  72 .TP
  73 \(bu
  74 perform a "make lint" of the kernel and report errors
  75 .TP
  76 \(bu
  77 perform a "make check" to report hdrchk/cstyle errors
  78 .TP
  79 \(bu
  80 report the presence of any core files
  81 .TP
  82 \(bu
  83 check the ELF runtime attributes of all dynamic objects
  84 .TP
  85 \(bu
  86 check for unreferenced files
  87 .TP
  88 \(bu
  89 report on which proto area objects have changed (since the last build)
  90 .TP
  91 \(bu
  92 report the total build time
  93 .TP
  94 \(bu
  95 save a detailed log file for reference
  96 .TP
  97 \(bu
  98 mail the user a summary of the completed build
  99 .RE
 100 .LP
 101 The actions of the script are almost completely determined by
 102 the environment variables in the
 103 .I env
 104 file, the only necessary argument.  Ths only thing you really
 105 need to use
 106 .I nightly
 107 is an
 108 .I env
 109 file that does what you want.
 110 .LP
 111 Like most of the other build tools in usr/src/tools, this script tends
 112 to change on a fairly regular basis; do not expect to be able to build
 113 OS/Net with a version of nightly significantly older than your source
 114 tree.  It has what is effectively a Consolidation Private relationship
 115 to other build tools and with many parts of the OS/Net makefiles,
 116 although it may also be used to build other consolidations.
 117 .LP
 118 .SH NIGHTLY_OPTIONS
 119 The environment variable NIGHTLY_OPTIONS controls the actions
 120 .I nightly
 121 will take as it proceeds.
 122 The -i, -n, +t and -V options may also be used from the command
 123 line to control the actions without editing your environment file.
 124 The -i and -n options complete the build more quickly by bypassing
 125 some actions. If NIGHTLY_OPTIONS is not set, then "-Bmt" build
 126 options will be used.
```

```
.B Basic action options
.TP 10
.B \-D
Do a build with DEBUG on (non-DEBUG is built by default)
.TP
.B \-F
Do _not_ do a non-DEBUG build (use with -D to get just a DEBUG build)
.TP
.B \-M
Do not run pmodes (safe file permission checker)
.TP
.B \-i
Do an incremental build, suppressing the "make clobber" that by
default removes all existing binaries and derived files.  From the
command line, -i also suppresses the lint pass and the cstyle/hdrchk
pass
.TP
.B \-n
Suppress the bringover so that the build will start immediately with
current source code
.TP
.B \-o
Do an "old style" (pre-S10) build using root privileges to set OWNER
and GROUP from the Makefiles.
.TP
.B \-p
Create packages for regular install
.TP
.B \-U
Update proto area in the parent workspace
.TP
.B \-u
Update the parent workspace with files generated by the build, as follows.
.RS
.TP
\(bu
Copy proto_list_${MACH} and friends to usr/src in the parent.
.TP
\(bu
When used with -f, build a usr/src/unrefmaster.out in
the parent by merging all the usr/src/unref-${MACH}.out files in the
parent.
.TP
\(bu
When used with -A or -r, copy the contents of the resulting
ELF-data.${MACH} directory to usr/src/ELF-data.${MACH} in the parent
workspace.
.RE
.TP
.B \-m
Send mail to $MAILTO at end of build
.TP
.B \-t
Build and use the tools in $SRC/tools (default setting).
.TP
.B \+t
Use the build tools in "$ONBLD_TOOLS/bin".

.LP
.B Code checking options
.TP 10
.B \-A
Check for ABI discrepancies in .so files.
It is only required for shared object developers when there is an
addition, deletion or change of interface in the .so files.
.TP
```



```
.B \-C
Check for cstyle/hdrchk errors
.TP
.B \-f
Check for unreferenced files.  Since the full workspace must be built
in order to accurately identify unreferenced files, -f is ignored for
incremental (-i) builds, or builds that do not include -l, and -p.
.TP
.B \-r
Check the ELF runtime attributes of all dynamic objects
.TP
.B \-l
Do "make lint" in $LINTDIRS (default: $SRC n)
.TP
.B \-N
Do not run protocmp or checkpaths (note: this option is not
recommended, especially in conjunction with the \-p option)
.TP
.B \-W
Do not report warnings (for freeware gate ONLY)
.TP
.B \-w
Report which proto area objects differ between this and the last build.
See wsdiff(1) for details. Note that the proto areas used for comparison
are the last ones constructed as part of the build. As an example, if both
a non-debug and debug build are performed (in that order), then the debug
proto area will be used for comparison (which might not be what you want).
.LP
.B Groups of options
.TP 10
.B \-G
Gate keeper default group of options (-u)
.TP
.B \-I
Integration engineer default group of options (-mpu)
.TP
.B \-R
Default group of options for building a release (-mp)

.LP
.B Source Build options
.TP 10
.B \-S E | D | H
Build the Export, Domestic, or Hybrid source product. Only Export and
Domestic are truly buildable at this time.
.TP 10
.B \-S O
Simulate an OpenSolaris build on a full tree.  This can be used by
internal developers to ensure that they haven’t broken the build for
external developers.
.LP
Source build options only make sense for a full internal tree (open
and closed source).  Only one source build option can be specified at
a time.

.LP
.B Miscellaneous options
.TP 10
.B \-O
generate deliverables for OpenSolaris.  Tarballs containing signed
cryptographic binaries and binaries
of closed-source components are put in $CODEMGR_WS.  (The
cryptographic tarballs are copies of the
ones that are put in the parent directory of
$PKGARCHIVE.)
.TP 10
```

```
.B \-V VERS
set the build version string to VERS, overriding VERSION
.TP
.B \-X
Copies the proto area and packages from the IHV and IHV-bin gates into the
nightly proto and package areas.  This is only available on i386.  See
.B REALMODE ENVIRONMENT VARIABLES
and
.B BUILDING THE IHV WORKSPACE
below.

.LP
.SH ENVIRONMENT VARIABLES
.LP
Here is a list of prominent environment variables that
.I nightly
references and the meaning of each variable.
.LP
.RE
.B CODEMGR_WS
.RS 5
The root of your workspace, including whatever metadata is kept by
the source code management system.  This is the workspace in which the
build will be done.
.RE
.LP
.B PARENT_WS
.RS 5
The root of the workspace that is the parent of the
one being built.  This is particularly relevant for configurations
with a main
workspace and build workspaces underneath it; see the
\-u and \-U
options as well as the PKGARCHIVE environment variable, for more
information.
.RE
.LP
.B BRINGOVER_WS
.RS 5
This is the workspace from which
.I nightly
will fetch sources to either populate or update your workspace;
it defaults to $CLONE_WS.
.RE
.LP
.B CLOSED_BRINGOVER_WS
.RS 5
A full Mercurial workspace has two repositories: one for open source
and one for closed source.  If this variable is non-null,
.I nightly
will pull from the repository that it names to get the closed source.
It defaults to $CLOSED_CLONE_WS.
.LP
If $CODEMGR_WS already exists and contains only the open repository,
.I nightly
will ignore this variable; you'll need to pull the closed repository
by hand if you want it.
.RE
.LP
.B CLONE_WS
.RS 5
This is the workspace from which
.I nightly
will fetch sources by default.  This is
often distinct from the parent, particularly if the parent is a gate.
.RE
```



```
.LP
.B CLOSED_CLONE_WS
.RS 5
This is the default closed-source Mercurial repository that
.I nightly
might pull from (see
.B CLOSED_BRINGOVER_WS
for details).
.RE
.LP
.B SRC
.RS 5
Root of OS-Net source code, referenced by the Makefiles.  It is
the starting point of build activity.  It should be expressed
in terms of $CODEMGR_WS.
.RE
.LP
.B ROOT
.RS 5
Root of the proto area for the build.  The makefiles direct
installation of build products to this area and
direct references to these files by builds of commands and other
targets.  It should be expressed in terms of $CODEMGR_WS.
.LP
If $MULTI_PROTO is "no", $ROOT may contain a DEBUG or non-DEBUG
build.  If $MULTI_PROTO is "yes", $ROOT contains the DEBUG build and
$ROOT-nd contains the non-DEBUG build.
.LP
For OpenSolaris deliveries (\fB\-O\fR), $ROOT-closed contains a parallel
proto area containing the DEBUG build of just usr/closed components, and
$ROOT-nd-closed contains the non-DEBUG equivalent.
.RE
.LP
.B TOOLS_ROOT
.RS 5
Root of the tools proto area for the build.  The makefiles direct
installation of tools build products to this area.  Unless \fB+t\fR
is part of $NIGHTLY_OPTIONS, these tools will be used during the
build.
.LP
As built by nightly, this will always contain non-DEBUG objects.
Therefore, this will always have a -nd suffix, regardless of
$MULTI_PROTO.
.RE
.LP
.B MACH
.RS 5
The instruction set architecture of the build machine as given
by \fIuname -p\fP, e.g. sparc, i386.
.RE
.LP
.B LOCKNAME
.RS 5
The name of the file used to lock out multiple runs of
.IR nightly .
This should generally be left to the default setting.
.RE
.LP
.B ATLOG
.RS 5
The location of the log directory maintained by
.IR nightly .
This should generally be left to the default setting.
.RE
.LP
.B LOGFILE
```

```
.RS 5
The name of the log file in the $ATLOG directory maintained by
.IR nightly .
This should generally be left to the default setting.
.RE
.LP
.B STAFFER
.RS 5
The non-root account to use on the build machine for the
bringover from the clone or parent workspace.
This may not be the same identify used by the SCM.
.RE
.LP
.B MAILTO
.RS 5
The address to be used to send completion e-mail at the end of
the build (for the \-m option).
.RE
.LP
.B MAILFROM
.RS 5
The address to be used for From: in the completion e-mail at the
end of the build (for the \-m option).
.RE
.LP
.B REF_PROTO_LIST
.RS 5
Name of file used with protocmp to compare proto area contents.
.RE
.LP
.B PARENT_ROOT
.RS 5
The parent root, which is the destination for copying the proto
area(s) when using the \-U option.
.RE
.LP
.B PARENT_TOOLS_ROOT
.RS 5
The parent tools root, which is the destination for copying the tools
proto area when using the \-U option.
.RE
.LP
.B RELEASE
.RS 5
The release version number to be used; e.g., 5.10.1 (Note: this is set
in Makefile.master and should not normally be overridden).
.RE
.LP
.B VERSION
.RS 5
The version text string to be used; e.g., "onnv:‘date ’+%Y-%m-%d’’".
.RE
.LP
.B RELEASE_DATE
.RS 5
The release date text to be used; e.g., October 2009. If not set in
your environment file, then this text defaults to the output from
$(LC_ALL=C date +"%B %Y"); e.g., "October 2009".
.RE
.LP
.B INTERNAL_RELEASE_BUILD
.RS 5
See Makefile.master - but it mostly controls id strings. Generally,
let
.I nightly
set this for you.
```



```
.RE
.LP
.B RELEASE_BUILD
.RS 5
Define this to build a release with a non-DEBUG kernel.
Generally, let
.I nightly
set this for you based on its options.
.RE
.LP
.B PKGARCHIVE
.RS 5
The destination for packages.  This may be relative to
$CODEMGR_WS for private packages or relative to $PARENT_WS
if you have different workspaces for different architectures
but want one hierarchy of packages.
.RE
.LP
.B MAKEFLAGS
.RS 5
Set default flags to make; e.g., -k to build all targets regardless of errors.
.RE
.LP
.B UT_NO_USAGE_TRACKING
.RS 5
Disables usage reporting by listed Devpro tools. Otherwise it sends mail
to some Devpro machine every time the tools are used.
.RE
.LP
.B LINTDIRS
.RS 5
Directories to lint with the \-l option.
.RE
.LP
.B BUILD_TOOLS
.RS 5
BUILD_TOOLS is the root of all tools including the compilers; e.g.,
/ws/onnv-tools.  It is used by the makefile system, but not nightly.
.RE
.LP
.B ONBLD_TOOLS
.RS 5
ONBLD_TOOLS is the root of all the tools that are part of SUNWonbld; e.g.,
/ws/onnv-tools/onbld.  By default, it is derived from
.BR BUILD_TOOLS .
It is used by the makefile system, but not nightly.
.RE
.LP
.B SPRO_ROOT
.RS 5
The gate-defined default location for the Sun compilers, e.g.
/ws/onnv-tools/SUNWspro.  By default, it is derived from
.BR BUILD_TOOLS .
It is used by the makefile system, but not nightly.
.RE
.LP
.B JAVA_ROOT
.RS 5
The location for the java compilers for the build, generally /usr/java.
.RE
.LP
.B OPTHOME
.RS 5
The gate-defined default location of things formerly in /opt; e.g.,
/ws/onnv-tools.  This is used by nightly, but not the makefiles.
.RE
```

```
.LP
.B TEAMWARE
.RS 5
The gate-defined default location for the Teamware tools; e.g.,
/ws/onnv-tools/SUNWspro.  By default, it is derived from
.BR OPTHOME .
This is used by nightly, but not the makefiles.  There is no
corresponding variable for Mercurial or Subversion, which are assumed
to be installed in the default path.
.RE
.LP
.B EXPORT_SRC
.RS 5
The source product has no SCCS history, and is modified to remove source
that cannot be shipped. EXPORT_SRC is where the clear files are copied, then
modified with 'make EXPORT_SRC'.
.RE
.LP
.B CRYPT_SRC
.RS 5
CRYPT_SRC is similar to EXPORT_SRC, but after 'make CRYPT_SRC' the files in
xmod/cry_files are saved. They are dropped on the exportable source to create
the domestic build.
.LP
.RE
.B OPEN_SRCDIR
.RS 5
The open source tree is copied to this directory when simulating an
OpenSolaris build (\fB\-S O\fR).  It defaults to $CODEMGR_WS/open_src.
.LP
.RE
.B ON_CLOSED_BINS
.RS 5
OpenSolaris builds do not contain the closed source tree.  Instead,
the developer downloads a closed binaries tree and unpacks it.
.B ON_CLOSED_BINS
tells nightly
where to find these closed binaries, so that it can add them into the
build.
.LP
.RE
.B ON_CRYPTO_BINS
.RS 5
This is the path to a compressed tarball that contains debug
cryptographic binaries that have been signed to allow execution
outside of Sun, e.g., $PARENT_WS/packages/$MACH/on-crypto.$MACH.bz2.
.I nightly
will automatically adjust the path for non-debug builds.  This tarball
is needed if the closed-source tree is not present.  Also, it is
usually needed when generating OpenSolaris deliverables from a project
workspace.  This is because most projects do not have access to the
necessary key and certificate that would let them sign their own
cryptographic binaries.
.LP
.RE
.B CHECK_PATHS
.RS 5
Normally, nightly runs the 'checkpaths' script to check for
discrepancies among the files that list paths to other files, such as
exception lists and req.flg.  Set this flag to 'n' to disable this
check, which appears in the nightly output as "Check lists of files."
.RE
.LP
.B CHECK_DMAKE
.RS 5
Nightly validates that the version of dmake encountered is known to be
```



```
safe to use.  Set this flag to 'n' to disable this test, allowing any
version of dmake to be used.
.RE
.LP
.B MULTI_PROTO
.RS 5
If "no" (the default),
.I nightly
will reuse $ROOT for both the DEBUG and non-DEBUG builds.  If "yes",
the DEBUG build will go in $ROOT and the non-DEBUG build will go in
$ROOT-nd.  Other values will be treated as "no".  Use of the
.B \-O
flag forces MULTI_PROTO to "yes".
.RE
.LP
.SH NIGHTLY HOOK ENVIRONMENT VARIABLES
.LP
Several optional environment variables may specify commands to run at
various points during the build.  Commands specified in the hook
variable will be run in a subshell; command output will be appended to
the mail message and log file.  If the hook exits with a non-zero
status, the build is aborted immediately.  Environment variables
defined in the environment file will be available.
.LP
.B SYS_PRE_NIGHTLY
.RS 5
Run just after the workspace lock is acquired.  This is reserved for
per-build-machine customizations and should be set only in /etc/nightly.conf
.RE
.LP
.B PRE_NIGHTLY
.RS 5
Run just after SYS_PRE_NIGHTLY.
.RE
.LP
.B PRE_BRINGOVER
.RS 5
Run just before bringover is started; not run if no bringover is done.
.RE
.LP
.B POST_BRINGOVER
.RS 5
Run just after bringover completes; not run if no bringover is done.
.RE
.LP
.B POST_NIGHTLY
.RS 5
Run after the build completes, with the return status of nightly - one
of "Completed", "Interrupted", or "Failed" - available in the
environment variable NIGHTLY_STATUS.
.RE
.LP
.B SYS_POST_NIGHTLY
.RS 5
This is reserved for per-build-machine customizations, and runs
immedately after POST_NIGHTLY.
.RE
.LP
.SH REALMODE ENVIRONMENT VARIABLES
.LP
The following environment variables referenced by
.I nightly
are only required when the -X option is used.
.LP
.RE
.B IA32_IHV_WS
```

```
.RS 5
Reference to the IHV workspace containing IHV driver binaries.
The IHV workspace must be fully built before starting the ON realmode build.
.LP
.RE
.B IA32_IHV_ROOT
.RS 5
Reference to the IHV workspace proto area.
The IHV workspace must be fully built before starting the ON realmode build.
.LP
.RE
.B IA32_IHV_PKGS
.RS 5
Reference to the IHV workspace packages.  If this is empty or the directory
is non-existent, then nightly will skip copying the packages.
.LP
.RE
.B IA32_IHV_BINARY_PKGS
.RS 5
Reference to binary-only IHV packages.  If this is empty or the directory
is non-existent, then nightly will skip copying the packages.
.LP
.RE
.B SPARC_RM_PKGARCHIVE
.RS 5
Destination for sparc realmode package SUNWrmodu.
Yes, this sparc package really is built on x86.
.SH FILES
.LP
.RS 5
/etc/nightly.conf
.RE
.LP
If present, nightly executes this file just prior to executing the
.I env
file.
.SH BUILDING THE IHV WORKSPACE
.LP
The IHV workspace can be built with
.I nightly.
The recommended options are:
.LP
.RS 5
NIGHTLY_OPTIONS="-pmWN"
.RE
.LP
None of the realmode environment variables needed for ON realmode builds
are required to build the IHV workspace.
.SH EXAMPLES
.LP
Start with the example file in usr/src/tools/env/developer.sh
(or gatekeeper.sh), copy to myenv and make your changes.
.LP
.PD 0
# grep NIGHTLY_OPTIONS myenv
.LP
NIGHTLY_OPTIONS="-ACrlapDm"
.LP
export NIGHTLY_OPTIONS
.LP
# /opt/onbld/bin/nightly -i myenv
.PD
.LP
.SH SEE ALSO
.BR bldenv (1)
```

```
**********************************************************
   85819 Thu Jul 11 01:29:46 2013
new/usr/src/tools/scripts/nightly.sh
first pass
**********************************************************
```

_____*unchanged_portion_omitted_*

```
159 #
160 # usage: filelist DESTDIR PATTERN
161 #
162 function filelist {
163         DEST=$1
164         PATTERN=$2
165         cd ${DEST}

167         OBJFILES=${ORIG_SRC}/xmod/obj_files
168         if [ ! -f ${OBJFILES} ]; then
169                 return;
170         fi
171         for i in `grep -v '^#' ${OBJFILES} | \
172             grep ${PATTERN} | cut -d: -f2 | tr -d ' \t'`
173         do
174                 # wildcard expansion
175                 for j in $i
176                 do
177                         if [ -f "$j" ]; then
178                                 echo $j
179                         fi
180                         if [ -d "$j" ]; then
181                                 echo $j
182                         fi
183                 done
184         done | sort | uniq
166 }
```

_____*unchanged_portion_omitted_*

```
365 #
366 # function to create (but not build) the export/crypt source tree.
367 # usage: set_up_source_build CODEMGR_WS DESTDIR MAKE_TARGET
368 # Sets SRC to the modified source tree, for use by the caller when it
369 # builds the tree.
370 #
371 function set_up_source_build {
372         WS=$1
373         DEST=$2
374         MAKETARG=$3

376         copy_source $WS $DEST $MAKETARG usr
377         if (( $? != 0 )); then
378             echo "\nCould not copy source tree for source build." |
379                 tee -a $mail_msg_file >> $LOGFILE
380             build_ok=n
381             return
382         fi

384         SRC=${DEST}/usr/src

386         cd $SRC
387         rm -f ${MAKETARG}.out
388         echo "making ${MAKETARG} in ${SRC}." >> $LOGFILE
389         /bin/time $MAKE -e ${MAKETARG} 2>&1 | \
390             tee -a $SRC/${MAKETARG}.out >> $LOGFILE
391         echo "\n==== ${MAKETARG} build errors ====\n" >> $mail_msg_file
392         egrep ":" $SRC/${MAKETARG}.out | \
393                 egrep -e "(^${MAKE}:|[  ]error[:        \n])" | \
394                 egrep -v "Ignoring unknown host" | \
```



```
395                 egrep -v "warning" >> $mail_msg_file

397         echo "clearing state files." >> $LOGFILE
398         find . -name '.make*' -exec rm -f {} \;

419         cd ${DEST}
420         if [ "${MAKETARG}" = "CRYPT_SRC" ]; then
421                 rm -f ${CODEMGR_WS}/crypt_files.cpio.Z
422                 echo "\n==== xmod/cry_files that don't exist ====\n" | \
423                     tee -a $mail_msg_file >> $LOGFILE
424                 CRYPT_FILES=${WS}/usr/src/xmod/cry_files
425                 for i in `cat ${CRYPT_FILES}`
426                 do
427                         # make sure the files exist
428                         if [ -f "$i" ]; then
429                                 continue
430                         fi
431                         if [ -d "$i" ]; then
432                                 continue
433                         fi
434                         echo "$i" | tee -a $mail_msg_file >> $LOGFILE
435                 done
436                 find `cat ${CRYPT_FILES}` -print 2>/dev/null | \
437                     cpio -ocB 2>/dev/null | \
438                     compress > ${CODEMGR_WS}/crypt_files.cpio.Z
439         fi

441         if [ "${MAKETARG}" = "EXPORT_SRC" ]; then
442                 # rename first, since we might restore a file
443                 # of the same name (mapfiles)
444                 rename_files ${EXPORT_SRC} EXPORT_SRC
445                 if [ "$SH_FLAG" = "y" ]; then
446                         hybridize_files ${EXPORT_SRC} EXPORT_SRC
447                 fi
448         fi

450         # save the cleartext
451         echo "\n==== Creating ${MAKETARG}.cpio.Z ====\n" | \
452             tee -a $mail_msg_file >> $LOGFILE
453         cd ${DEST}
454         rm -f ${MAKETARG}.cpio.Z
455         find usr -depth -print | \
456             grep -v usr/src/${MAKETARG}.out | \
457             cpio -ocB 2>/dev/null | \
458             compress > ${CODEMGR_WS}/${MAKETARG}.cpio.Z
459         if [ "${MAKETARG}" = "EXPORT_SRC" ]; then
460                 restore_binaries ${EXPORT_SRC} EXPORT_SRC
461         fi

463         if [ "${MAKETARG}" = "CRYPT_SRC" ]; then
464                 restore_binaries ${CRYPT_SRC} CRYPT_SRC
465         fi

399 }
```

_____*unchanged_portion_omitted_*

```
966 MACH=`uname -p`

968 if [ "$OPTHOME" = "" ]; then
969         OPTHOME=/opt
970         export OPTHOME
971 fi
972 if [ "$TEAMWARE" = "" ]; then
973         TEAMWARE=$OPTHOME/teamware
974         export TEAMWARE
```

```
 975 fi

 977 USAGE='Usage: nightly [-in] [+t] [-V VERS ] [ -S E|D|H|O ] <env_file>

 979 Where:
 980         -i      Fast incremental options (no clobber, lint, check)
 981         -n      Do not do a bringover
 982         +t      Use the build tools in $ONBLD_TOOLS/bin
 983         -V VERS set the build version string to VERS
 984         -S      Build a variant of the source product
 985                 E - build exportable source
 986                 D - build domestic source (exportable + crypt)
 987                 H - build hybrid source (binaries + deleted source)
 988                 O - build (only) open source

 990         <env_file>  file in Bourne shell syntax that sets and exports
 991         variables that configure the operation of this script and many of
 992         the scripts this one calls. If <env_file> does not exist,
 993         it will be looked for in $OPTHOME/onbld/env.

 995 non-DEBUG is the default build type. Build options can be set in the
 996 NIGHTLY_OPTIONS variable in the <env_file> as follows:

 998         -A      check for ABI differences in .so files
 999         -C      check for cstyle/hdrchk errors
1000         -D      do a build with DEBUG on
1001         -F      do _not_ do a non-DEBUG build
1002         -G      gate keeper default group of options (-au)
1003         -I      integration engineer default group of options (-ampu)
1004         -M      do not run pmodes (safe file permission checker)
1005         -N      do not run protocmp
1006         -O      generate OpenSolaris deliverables
1007         -R      default group of options for building a release (-mp)
1008         -U      update proto area in the parent
1009         -V VERS set the build version string to VERS
1010         -X      copy x86 IHV proto area
1011         -f      find unreferenced files
1012         -i      do an incremental build (no "make clobber")
1013         -l      do "make lint" in $LINTDIRS (default: $SRC y)
1014         -m      send mail to $MAILTO at end of build
1015         -n      do not do a bringover
1016         -o      build using root privileges to set OWNER/GROUP (old style)
1017         -p      create packages
1018         -r      check ELF runtime attributes in the proto area
1019         -t      build and use the tools in $SRC/tools (default setting)
1020         +t      Use the build tools in $ONBLD_TOOLS/bin
1021         -u      update proto_list_$MACH and friends in the parent workspace;
1022                 when used with -f, also build an unrefmaster.out in the parent
1023         -w      report on differences between previous and current proto areas
1024         -z      compress cpio archives with gzip
1025         -W      Do not report warnings (freeware gate ONLY)
1026         -S      Build a variant of the source product
1027                 E - build exportable source
1028                 D - build domestic source (exportable + crypt)
1029                 H - build hybrid source (binaries + deleted source)
1030                 O - build (only) open source
1031 '
1032 #
1101 #       -x      less public handling of xmod source for the source product
1102 #
1033 #       A log file will be generated under the name $LOGFILE
1034 #       for partially completed build and log.`date '+%F'`
1035 #       in the same directory for fully completed builds.
1036 #

1038 # default values for low-level FLAGS; G I R are group FLAGS
```



```
1039 A_FLAG=n
1040 C_FLAG=n
1041 D_FLAG=n
1042 F_FLAG=n
1043 f_FLAG=n
1044 i_FLAG=n; i_CMD_LINE_FLAG=n
1045 l_FLAG=n
1046 M_FLAG=n
1047 m_FLAG=n
1048 N_FLAG=n
1049 n_FLAG=n
1050 O_FLAG=n
1051 o_FLAG=n
1052 P_FLAG=n
1053 p_FLAG=n
1054 r_FLAG=n
1055 T_FLAG=n
1056 t_FLAG=y
1057 U_FLAG=n
1058 u_FLAG=n
1059 V_FLAG=n
1060 W_FLAG=n
1061 w_FLAG=n
1062 X_FLAG=n
1063 SD_FLAG=n
1064 SE_FLAG=n
1065 SH_FLAG=n
1066 SO_FLAG=n
1067 #
1068 XMOD_OPT=
1069 #
1070 build_ok=y

1072 function is_source_build {
1073         [ "$SE_FLAG" = "y" -o "$SD_FLAG" = "y" -o \
1074           "$SH_FLAG" = "y" -o "$SO_FLAG" = "y" ]
1075         return $?
1076 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1745 Thu Jul 11 01:29:46 2013
new/usr/src/tools/scripts/stdenv.sh
first pass
**********************************************************
```

```
#
# 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 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# ident "%Z%%M% %I%     %E% SMI"
#

#
# Shell script fragment to set standard build environment variables,
# for use by bldenv(1) and nightly(1).  Can be overridden by the
# user's environment file.  Because bldenv and nightly are both ksh
# scripts, we can use ksh syntax here.
#

#
# the source product has no SCCS history, and is modified to remove source
# that cannot be shipped. EXPORT_SRC is where the clear files are copied, then
# modified with 'make EXPORT_SRC'.
#
[ -n "$EXPORT_SRC" ] || export EXPORT_SRC="$CODEMGR_WS/export_src"

#
# CRYPT_SRC is similar to EXPORT_SRC, but after 'make CRYPT_SRC' the files in
# xmod/cry_files are saved. They are dropped on the exportable source to create
# the domestic build.
#
[ -n "$CRYPT_SRC" ] || export CRYPT_SRC="$CODEMGR_WS/crypt_src"

#
# OPEN_SRCDIR is where we copy the open tree to so that we can be sure
# we don't have a hidden dependency on closed code.  The name ends in
# "DIR" to avoid confusion with the flags related to open source
# builds.
#
[ -n "$OPEN_SRCDIR" ] || export OPEN_SRCDIR="$CODEMGR_WS/open_src"

#
# Flag to enable creation of per-build-type proto areas.  If "yes",
# more proto areas are created, which speeds up incremental builds but
# increases storage consumption.  Will be forced to "yes" for
```



```
# OpenSolaris deliveries.
#
[ -n "$MULTI_PROTO" ] || export MULTI_PROTO=no
```

```
**********************************************************
    6908 Thu Jul 11 01:29:47 2013
new/usr/src/uts/Makefile
onc plus-be-gone
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
  23 #
  24 # include global definitions
  25 include ../Makefile.master

  27 #
  28 # List of architectures to build as part of the standard build.
  29 #
  30 # Some of these architectures are built in parallel (see i386_PARALLEL and
  31 # sparc_PARALLEL). This requires building some parts first before parallel build
  32 # can start. Platform make files know what should be built as a prerequisite for
  33 # the parallel build to work. The i386_PREREQ and sparc_PREREQ variables tell
  34 # which platform directory to enter to start making prerequisite dependencies.
  35 #
  36 sparc_ARCHITECTURES = sun4v sun4u sparc

  38 i386_ARCHITECTURES = i86pc i86xpv intel

  40 #
  41 # For i386 all architectures can be compiled in parallel.
  42 #
  43 # intel/Makefile knows how to build prerequisites needed for parallel build.
  44 #
  45 i386_PREREQ = intel
  46 i386_PARALLEL = $(i386_ARCHITECTURES)

  48 #
  49 # For sparc all architectures can be compiled in parallel.
  50 #
  51 # sun4/Makefile knows how to build prerequisites needed for parallel build.
  52 # can start.
  53 #
  54 sparc_PREREQ = sun4
  55 sparc_PARALLEL = $(sparc_ARCHITECTURES)

  57 #
  58 # Platforms defined in $(MACH)_PARALLEL are built in parallel. DUMMY is placed
  59 # at the end in case $(MACH)_PARALLEL is empty to prevent everything going in
  60 # parallel.
  61 #
```



```
  62 .PARALLEL: $($(MACH)_PARALLEL) DUMMY

  64 #
  65 # For build prerequisites we use a special target which is constructed by adding
  66 # '.prereq' suffix to the $(MACH)_PREREQ.
  67 #
  68 PREREQ_TARGET = $($(MACH)_PREREQ:%=%.prereq)


  71 def             :=      TARGET= def
  72 all             :=      TARGET= all
  73 install         :=      TARGET= install
  74 install_h       :=      TARGET= install_h
  75 clean           :=      TARGET= clean
  76 clobber         :=      TARGET= clobber
  77 lint            :=      TARGET= lint
  78 clean.lint      :=      TARGET= clean.lint
  79 check           :=      TARGET= check
  80 modlist         :=      TARGET= modlist
  81 modlist         :=      NO_STATE= -K $$MODSTATE$$$$

  83 .KEEP_STATE:

  85 def all lint: all_h $(PMTMO_FILE) $($(MACH)_ARCHITECTURES)

  87 install: all_h install_dirs $(PMTMO_FILE) $($(MACH)_ARCHITECTURES)

  89 install_dirs:
  90         @cd ..; pwd; $(MAKE) rootdirs
  91         @pwd

  93 #
  94 # Rule to build prerequisites. The left part of the pattern will match
  95 # PREREQ_TARGET.
  96 #
  97 # The location of the Makefile is determined by strippinng '.prereq' suffix from
  98 # the target name. We add '.prereq' suffix to the target passed to the child
  99 # Makefile so that it can distinguish prerequisite build from the regular one.
 100 #
 101 #
 102 %.prereq:
 103         @cd $(@:%.prereq=%); pwd; $(MAKE) $(NO_STATE) $(TARGET).prereq

 105 #
 106 # Rule to build architecture files. Build all required prerequisites and then
 107 # build the rest (potentially in parallel).
 108 #
 109 $($(MACH)_ARCHITECTURES): $(PREREQ_TARGET) FRC
 110         @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

 112 $(PMTMO_FILE) pmtmo_file: $(PATCH_MAKEUP_TABLE)
 113         @if [ -z "$(PATCH_MAKEUP_TABLE)" ] ; then \
 114                 echo 'ERROR: $$(PATCH_MAKEUP_TABLE) not set' \
 115                         'in environment' >&2 ; \
 116                 exit 1 ; \
 117         fi
 118         RELEASE="$(RELEASE)" MACH="$(MACH)" \
 119             $(CTFCVTPTBL) -o $(PMTMO_FILE) $(PATCH_MAKEUP_TABLE)

 121 #
 122 # The following is the list of directories which contain Makefiles with
 123 # targets to install header file. The machine independent headers are
 124 # installed by invoking the Makefile in the directory containing the
 125 # header files. Machine and architecture dependent headers are installed
 126 # by invoking the main makefile for that architecture/machine which,
 127 # in turn, is responsible for invoking the Makefiles which install headers.
```

```
128 # It is done this way so as not to assume that all of the header files in
129 # the architecture/machine dependent subdirectories are in completely
130 # isomorphic locations.
131 #
132 COMMON_HDRDIRS= common/avs \
133                 common/c2 \
134                 common/des \
135                 common/fs \
136                 common/gssapi \
137                 common/idmap \
138                 common/inet \
139                 common/inet/ipf/netinet \
140                 common/inet/kssl \
141                 common/inet/nca \
142                 common/inet/sockmods/netpacket \
143                 common/io/bpf/net \
144                 common/ipp \
145                 common/net \
146                 common/netinet \
147                 common/nfs \
148                 common/pcmcia/sys \
149                 common/rpc \
150                 common/rpcsvc \
151                 common/sharefs \
152                 common/smb \
153                 common/smbsrv \
154                 common/sys \
155                 common/vm


158 # These aren't the only headers in closed.  But the other directories
159 # are simple enough that they can be driven from the src tree.
160 $(CLOSED_BUILD)COMMON_HDRDIRS += $(CLOSED)/uts/common/sys

162 #
163 # Subset of COMMON_HDRDIRS in which at least one header is generated
164 # at runtime (e.g., rpcgen).  (This is a partial list; there are
165 # other directories that should be included and do not yet have the
166 # necessary Makefile support.  See 6414855.)
167 #
168 DYNHDRDIRS = common/rpcsvc common/idmap common/sys

170 sparc_HDRDIRS= sun/sys
171 i386_HDRDIRS= i86pc/vm i86xpv/vm

173 HDRDIRS= $(COMMON_HDRDIRS) $($(MACH)_HDRDIRS)
174 install_h check: $(HDRDIRS) $($(MACH)_ARCHITECTURES)

176 $(HDRDIRS): FRC
177         @cd $@; pwd; $(MAKE) $(TARGET)

179 # ensures that headers made by rpcgen and others are available in uts source
180 # for kernel builds to reference without building install_h
181 #
182 all_h: FRC
183         @cd common/sys; pwd; $(MAKE) $@
184         @cd common/rpc; pwd; $(MAKE) $@
185         @cd common/rpcsvc; pwd; $(MAKE) $@
186         @cd common/gssapi; pwd; $(MAKE) $@
187         @cd common/idmap; pwd; $(MAKE) $@

189 clean clobber: $($(MACH)_ARCHITECTURES) $(DYNHDRDIRS)
190         @if [ '$(PATCH_BUILD)' != '#' ] ; then \
191                 echo $(RM) $(PMTMO_FILE) ; \
192                 $(RM) $(PMTMO_FILE) ; \
193         fi
```



```
195 EXTRA_CLOBBER_TARGETS= common/avs/ns/rdc
196 clobber: $(EXTRA_CLOBBER_TARGETS)


199 clean.lint modlist: $($(MACH)_ARCHITECTURES)

201 ONC_FILES=      common/io/timod.c \
202                 common/os/sig.c \
203                 common/os/flock.c \
204                 common/os/sysent.c \
205                 common/os/swapgeneric.c \
206                 common/syscall/fcntl.c

208 # edit onc plus source files.
209 ONC_PLUS:       $(ONC_FILES:%=%_onc_plus)

201 #
202 # Cross-reference customization: build a cross-reference over all of
203 # the supported architectures.  Although there's no correct way to set
204 # the include path (since we don't know what architecture is the one
205 # the user will be interested in), it's historically been set to
206 # mirror the $(XRDIRS) list, and that works kinda sorta okay.
207 #
208 # We need to manually prune usr/closed/uts/{i86xpv|sfmmu|i86pc} since
209 # none of them exist.
210 #
211 SHARED_XRDIRS = $(sparc_ARCHITECTURES) $(i386_ARCHITECTURES) sun4 sfmmu \
212         sun common
213 CLOSED_XRDIRS = $(SHARED_XRDIRS:%=% ../../closed/uts/%)
214 XRDIRS = $(SHARED_XRDIRS)
215 CLOSED_XRDIRS_XEN = $(CLOSED_XRDIRS:../../closed/uts/i86xpv=)
216 CLOSED_XRDIRS_1 = $(CLOSED_XRDIRS_XEN:../../closed/uts/i86pc=)
217 $(CLOSED_BUILD)XRDIRS = $(CLOSED_XRDIRS_1:../../closed/uts/sfmmu=)

219 XRINCDIRS = $(XRDIRS)

221 cscope.out tags: FRC
222         $(XREF) -x $@

224 FRC:
```

```
**********************************************************
     972 Thu Jul 11 01:29:48 2013
new/usr/src/uts/common/Makefile
first pass
**********************************************************
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 #  You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #
  25 #
  26 # uts/common/Makefile
  27 #
  28 include $(SRC)/Makefile.master

  30 .KEEP_STATE:

  32 # EXPORT DELETE START
  33 # Special target to clean up the source tree for export distribution
  34 # Warning: This target changes the source tree
  35 EXPORT_SRC:
  36         $(RM) Makefile+ Makefile.rules+
  37         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
  38                 < Makefile > Makefile+
  39         $(MV) Makefile+ Makefile
  40         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
  41                 < Makefile.rules > Makefile.rules+
  42         $(MV) Makefile.rules+ Makefile.rules
  43         $(CHMOD) 444 Makefile Makefile.rules
  44 # EXPORT DELETE END
```

```
**********************************************************
   43366 Thu Jul 11 01:29:48 2013
new/usr/src/uts/common/Makefile.files
first pass
**********************************************************
#
# 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) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2012 Nexenta Systems, Inc. All rights reserved.
# Copyright (c) 2012 by Delphix. All rights reserved.
# Copyright (c) 2013 by Saso Kiselkov. All rights reserved.
#

#
# This Makefile defines all file modules for the directory uts/common
# and its children. These are the source files which may be considered
# common to all SunOS systems.

i386_CORE_OBJS += \
                atomic.o        \
                avintr.o        \
                pic.o

sparc_CORE_OBJS +=

COMMON_CORE_OBJS +=             \
                beep.o          \
                bitset.o        \
                bp_map.o        \
                brand.o         \
                cpucaps.o       \
                cmt.o           \
                cmt_policy.o    \
                cpu.o           \
                cpu_event.o     \
                cpu_intr.o      \
                cpu_pm.o        \
                cpupart.o       \
                cap_util.o      \
                disp.o          \
                group.o         \
                kstat_fr.o      \
                iscsiboot_prop.o        \
                lgrp.o          \
                lgrp_topo.o     \
                mmapobj.o       \
```



```
                mutex.o         \
                page_lock.o     \
                page_retire.o   \
                panic.o         \
                param.o         \
                pg.o            \
                pghw.o          \
                putnext.o       \
                rctl_proc.o     \
                rwlock.o        \
                seg_kmem.o      \
                softint.o       \
                string.o        \
                strtol.o        \
                strtoul.o       \
                strtoll.o       \
                strtoull.o      \
                thread_intr.o   \
                vm_page.o       \
                vm_pagelist.o   \
                zlib_obj.o      \
                clock_tick.o

CORE_OBJS +=    $(COMMON_CORE_OBJS) $($(MACH)_CORE_OBJS)

ZLIB_OBJS =     zutil.o zmod.o zmod_subr.o \
                adler32.o crc32.o deflate.o inffast.o \
                inflate.o inftrees.o trees.o

GENUNIX_OBJS += \
                access.o        \
                acl.o           \
                acl_common.o    \
                adjtime.o       \
                alarm.o         \
                aio_subr.o      \
                auditsys.o      \
                audit_core.o    \
                audit_zone.o    \
                audit_memory.o  \
                autoconf.o      \
                avl.o           \
                bdev_dsort.o    \
                bio.o           \
                bitmap.o        \
                blabel.o        \
                brandsys.o      \
                bz2blocksort.o  \
                bz2compress.o   \
                bz2decompress.o \
                bz2randtable.o  \
                bz2bzlib.o      \
                bz2crctable.o   \
                bz2huffman.o    \
                callb.o         \
                callout.o       \
                chdir.o         \
                chmod.o         \
                chown.o         \
                cladm.o         \
                class.o         \
                clock.o         \
                clock_highres.o \
                clock_realtime.o\
                close.o         \
                compress.o      \
```

```
128                 condvar.o       \
129                 conf.o          \
130                 console.o       \
131                 contract.o      \
132                 copyops.o       \
133                 core.o          \
134                 corectl.o       \
135                 cred.o          \
136                 cs_stubs.o      \
137                 dacf.o          \
138                 dacf_clnt.o     \
139                 damap.o \
140                 cyclic.o        \
141                 ddi.o           \
142                 ddifm.o         \
143                 ddi_hp_impl.o   \
144                 ddi_hp_ndi.o    \
145                 ddi_intr.o      \
146                 ddi_intr_impl.o \
147                 ddi_intr_irm.o  \
148                 ddi_nodeid.o    \
149                 ddi_timer.o     \
150                 devcfg.o        \
151                 devcache.o      \
152                 device.o        \
153                 devid.o         \
154                 devid_cache.o   \
155                 devid_scsi.o    \
156                 devid_smp.o     \
157                 devpolicy.o     \
158                 disp_lock.o     \
159                 dnlc.o          \
160                 driver.o        \
161                 dumpsubr.o      \
162                 driver_lyr.o    \
163                 dtrace_subr.o   \
164                 errorq.o        \
165                 etheraddr.o     \
166                 evchannels.o    \
167                 exacct.o        \
168                 exacct_core.o   \
169                 exec.o          \
170                 exit.o          \
171                 fbio.o          \
172                 fcntl.o         \
173                 fdbuffer.o      \
174                 fdsync.o        \
175                 fem.o           \
176                 ffs.o           \
177                 fio.o           \
178                 flock.o         \
179                 fm.o            \
180                 fork.o          \
181                 vpm.o           \
182                 fs_reparse.o    \
183                 fs_subr.o       \
184                 fsflush.o       \
185                 ftrace.o        \
186                 getcwd.o        \
187                 getdents.o      \
188                 getloadavg.o    \
189                 getpagesizes.o  \
190                 getpid.o        \
191                 gfs.o           \
192                 rusagesys.o     \
193                 gid.o           \
```



```
194                 groups.o        \
195                 grow.o          \
196                 hat_refmod.o    \
197                 id32.o          \
198                 id_space.o      \
199                 inet_ntop.o     \
200                 instance.o      \
201                 ioctl.o         \
202                 ip_cksum.o      \
203                 issetugid.o     \
204                 ippconf.o       \
205                 kcpc.o          \
206                 kdi.o           \
207                 kiconv.o        \
208                 klpd.o          \
209                 kmem.o          \
210                 ksyms_snapshot.o        \
211                 l_strplumb.o    \
212                 labelsys.o      \
213                 link.o          \
214                 list.o          \
215                 lockstat_subr.o \
216                 log_sysevent.o  \
217                 logsubr.o       \
218                 lookup.o        \
219                 lseek.o         \
220                 ltos.o          \
221                 lwp.o           \
222                 lwp_create.o    \
223                 lwp_info.o      \
224                 lwp_self.o      \
225                 lwp_sobj.o      \
226                 lwp_timer.o     \
227                 lwpsys.o        \
228                 main.o          \
229                 mmapobjsys.o    \
230                 memcntl.o       \
231                 memstr.o        \
232                 lgrpsys.o       \
233                 mkdir.o         \
234                 mknod.o         \
235                 mount.o         \
236                 move.o          \
237                 msacct.o        \
238                 multidata.o     \
239                 nbmlock.o       \
240                 ndifm.o         \
241                 nice.o          \
242                 netstack.o      \
243                 ntptime.o       \
244                 nvpair.o        \
245                 nvpair_alloc_system.o   \
246                 nvpair_alloc_fixed.o    \
247                 fnvpair.o       \
248                 octet.o         \
249                 open.o          \
250                 p_online.o      \
251                 pathconf.o      \
252                 pathname.o      \
253                 pause.o         \
254                 serializer.o    \
255                 pci_intr_lib.o  \
256                 pci_cap.o       \
257                 pcifm.o         \
258                 pgrp.o          \
259                 pgrpsys.o       \
```

```
260                 pid.o           \
261                 pkp_hash.o      \
262                 policy.o        \
263                 poll.o          \
264                 pool.o          \
265                 pool_pset.o     \
266                 port_subr.o     \
267                 ppriv.o         \
268                 printf.o        \
269                 priocntl.o      \
270                 priv.o          \
271                 priv_const.o    \
272                 proc.o          \
273                 procset.o       \
274                 processor_bind.o        \
275                 processor_info.o        \
276                 profil.o        \
277                 project.o       \
278                 qsort.o         \
279                 rctl.o          \
280                 rctlsys.o       \
281                 readlink.o      \
282                 refstr.o        \
283                 rename.o        \
284                 resolvepath.o   \
285                 retire_store.o  \
286                 process.o       \
287                 rlimit.o        \
288                 rmap.o          \
289                 rw.o            \
290                 rwstlock.o      \
291                 sad_conf.o      \
292                 sid.o           \
293                 sidsys.o        \
294                 sched.o         \
295                 schedctl.o      \
296                 sctp_crc32.o    \
297                 seg_dev.o       \
298                 seg_kp.o        \
299                 seg_kpm.o       \
300                 seg_map.o       \
301                 seg_vn.o        \
302                 seg_spt.o       \
303                 semaphore.o     \
304                 sendfile.o      \
305                 session.o       \
306                 share.o         \
307                 shuttle.o       \
308                 sig.o           \
309                 sigaction.o     \
310                 sigaltstack.o   \
311                 signotify.o     \
312                 sigpending.o    \
313                 sigprocmask.o   \
314                 sigqueue.o      \
315                 sigsendset.o    \
316                 sigsuspend.o    \
317                 sigtimedwait.o  \
318                 sleepq.o        \
319                 sock_conf.o     \
320                 space.o         \
321                 sscanf.o        \
322                 stat.o          \
323                 statfs.o        \
324                 statvfs.o       \
325                 stol.o          \
```



```
326                 str_conf.o      \
327                 strcalls.o      \
328                 stream.o        \
329                 streamio.o      \
330                 strext.o        \
331                 strsubr.o       \
332                 strsun.o        \
333                 subr.o          \
334                 sunddi.o        \
335                 sunmdi.o        \
336                 sunndi.o        \
337                 sunpci.o        \
338                 sunpm.o         \
339                 sundlpi.o       \
340                 suntpi.o        \
341                 swap_subr.o     \
342                 swap_vnops.o    \
343                 symlink.o       \
344                 sync.o          \
345                 sysclass.o      \
346                 sysconfig.o     \
347                 sysent.o        \
348                 sysfs.o         \
349                 systeminfo.o    \
350                 task.o          \
351                 taskq.o         \
352                 tasksys.o       \
353                 time.o          \
354                 timer.o         \
355                 times.o         \
356                 timers.o        \
357                 thread.o        \
358                 tlabel.o        \
359                 tnf_res.o       \
360                 turnstile.o     \
361                 tty_common.o    \
362                 u8_textprep.o   \
363                 uadmin.o        \
364                 uconv.o         \
365                 ucredsys.o      \
366                 uid.o           \
367                 umask.o         \
368                 umount.o        \
369                 uname.o         \
370                 unix_bb.o       \
371                 unlink.o        \
372                 urw.o           \
373                 utime.o         \
374                 utssys.o        \
375                 uucopy.o        \
376                 vfs.o           \
377                 vfs_conf.o      \
378                 vmem.o          \
379                 vm_anon.o       \
380                 vm_as.o         \
381                 vm_meter.o      \
382                 vm_pageout.o    \
383                 vm_pvn.o        \
384                 vm_rm.o         \
385                 vm_seg.o        \
386                 vm_subr.o       \
387                 vm_swap.o       \
388                 vm_usage.o      \
389                 vnode.o         \
390                 vuid_queue.o    \
391                 vuid_store.o    \
```

```
392                 waitq.o         \
393                 watchpoint.o    \
394                 yield.o         \
395                 scsi_confdata.o \
396                 xattr.o         \
397                 xattr_common.o  \
398                 xdr_mblk.o      \
399                 xdr_mem.o       \
400                 xdr.o           \
401                 xdr_array.o     \
402                 xdr_refer.o     \
403                 xhat.o          \
404                 zone.o

406 #
407 #       Stubs for the stand-alone linker/loader
408 #
409 sparc_GENSTUBS_OBJS =   \
410         kobj_stubs.o

412 i386_GENSTUBS_OBJS =

414 COMMON_GENSTUBS_OBJS =

416 GENSTUBS_OBJS += $(COMMON_GENSTUBS_OBJS) $($(MACH)_GENSTUBS_OBJS)

418 #
419 #       DTrace and DTrace Providers
420 #
421 DTRACE_OBJS += dtrace.o dtrace_isa.o dtrace_asm.o

423 SDT_OBJS += sdt_subr.o

425 PROFILE_OBJS += profile.o

427 SYSTRACE_OBJS += systrace.o

429 LOCKSTAT_OBJS += lockstat.o

431 FASTTRAP_OBJS += fasttrap.o fasttrap_isa.o

433 DCPC_OBJS += dcpc.o

435 #
436 #       Driver (pseudo-driver) Modules
437 #
438 IPP_OBJS +=     ippctl.o

440 AUDIO_OBJS += audio_client.o audio_ddi.o audio_engine.o \
441             audio_fltdata.o audio_format.o audio_ctrl.o \
442             audio_grc3.o audio_output.o audio_input.o \
443             audio_oss.o audio_sun.o

445 AUDIOEMU10K_OBJS += audioemu10k.o

447 AUDIOENS_OBJS += audioens.o

449 AUDIOVIA823X_OBJS += audiovia823x.o

451 AUDIOVIA97_OBJS += audiovia97.o

453 AUDIO1575_OBJS += audio1575.o

455 AUDIO810_OBJS += audio810.o

457 AUDIOCMI_OBJS += audiocmi.o
```



```
459 AUDIOCMIHD_OBJS += audiocmihd.o

461 AUDIOHD_OBJS += audiohd.o

463 AUDIOIXP_OBJS += audioixp.o

465 AUDIOLS_OBJS += audiols.o

467 AUDIOP16X_OBJS += audiop16x.o

469 AUDIOPCI_OBJS += audiopci.o

471 AUDIOSOLO_OBJS += audiosolo.o

473 AUDIOTS_OBJS += audiots.o

475 AC97_OBJS += ac97.o ac97_ad.o ac97_alc.o ac97_cmi.o

477 BLKDEV_OBJS += blkdev.o

479 CARDBUS_OBJS += cardbus.o cardbus_hp.o cardbus_cfg.o

481 CONSKBD_OBJS += conskbd.o

483 CONSMS_OBJS +=  consms.o

485 OLDPTY_OBJS +=  tty_ptyconf.o

487 PTC_OBJS +=     tty_pty.o

489 PTSL_OBJS +=    tty_pts.o

491 PTM_OBJS +=     ptm.o

493 MII_OBJS +=     mii.o mii_cicada.o mii_natsemi.o mii_intel.o mii_qualsemi.o \
494                 mii_marvell.o mii_realtek.o mii_other.o

496 PTS_OBJS +=     pts.o

498 PTY_OBJS +=     ptms_conf.o

500 SAD_OBJS +=     sad.o

502 MD4_OBJS +=     md4.o md4_mod.o

504 MD5_OBJS +=     md5.o md5_mod.o

506 SHA1_OBJS +=    sha1.o sha1_mod.o

508 SHA2_OBJS +=    sha2.o sha2_mod.o

510 IPGPC_OBJS +=   classifierddi.o classifier.o filters.o trie.o table.o \
511                 ba_table.o

513 DSCPMK_OBJS +=  dscpmk.o dscpmkddi.o

515 DLCOSMK_OBJS += dlcosmk.o dlcosmkddi.o

517 FLOWACCT_OBJS +=        flowacctddi.o flowacct.o

519 TOKENMT_OBJS += tokenmt.o tokenmtddi.o

521 TSWTCL_OBJS +=  tswtcl.o tswtclddi.o

523 ARP_OBJS +=     arpddi.o
```

```
525 ICMP_OBJS +=    icmpddi.o

527 ICMP6_OBJS +=   icmp6ddi.o

529 RTS_OBJS +=     rtsddi.o

531 IP_ICMP_OBJS =  icmp.o icmp_opt_data.o
532 IP_RTS_OBJS =   rts.o rts_opt_data.o
533 IP_TCP_OBJS =   tcp.o tcp_fusion.o tcp_opt_data.o tcp_sack.o tcp_stats.o \
534                 tcp_misc.o tcp_timers.o tcp_time_wait.o tcp_tpi.o tcp_output.o \
535                 tcp_input.o tcp_socket.o tcp_bind.o tcp_cluster.o tcp_tunables.o
536 IP_UDP_OBJS =   udp.o udp_opt_data.o udp_tunables.o udp_stats.o
537 IP_SCTP_OBJS =  sctp.o sctp_opt_data.o sctp_output.o \
538                 sctp_init.o sctp_input.o sctp_cookie.o \
539                 sctp_conn.o sctp_error.o sctp_snmp.o \
540                 sctp_tunables.o sctp_shutdown.o sctp_common.o \
541                 sctp_timer.o sctp_heartbeat.o sctp_hash.o \
542                 sctp_bind.o sctp_notify.o sctp_asconf.o \
543                 sctp_addr.o tn_ipopt.o tnet.o ip_netinfo.o \
544                 sctp_misc.o
545 IP_ILB_OBJS =   ilb.o ilb_nat.o ilb_conn.o ilb_alg_hash.o ilb_alg_rr.o

547 IP_OBJS +=      igmp.o ipmp.o ip.o ip6.o ip6_asp.o ip6_if.o ip6_ire.o \
548                 ip6_rts.o ip_if.o ip_ire.o ip_listutils.o ip_mroute.o \
549                 ip_multi.o ip2mac.o ip_ndp.o ip_rts.o ip_srcid.o \
550                 ipddi.o ipdrop.o mi.o nd.o tunables.o optcom.o snmpcom.o \
551                 ipsec_loader.o spd.o ipclassifier.o inet_common.o ip_squeue.o \
552                 squeue.o ip_sadb.o ip_ftable.o proto_set.o radix.o ip_dummy.o \
553                 ip_helper_stream.o ip_tunables.o \
554                 ip_output.o ip_input.o ip6_input.o ip6_output.o ip_arp.o \
555                 conn_opt.o ip_attr.o ip_dce.o \
556                 $(IP_ICMP_OBJS) \
557                 $(IP_RTS_OBJS) \
558                 $(IP_TCP_OBJS) \
559                 $(IP_UDP_OBJS) \
560                 $(IP_SCTP_OBJS) \
561                 $(IP_ILB_OBJS)

563 IP6_OBJS +=     ip6ddi.o

565 HOOK_OBJS +=    hook.o

567 NETI_OBJS +=    neti_impl.o neti_mod.o neti_stack.o

569 KEYSOCK_OBJS += keysockddi.o keysock.o keysock_opt_data.o

571 IPNET_OBJS +=   ipnet.o ipnet_bpf.o

573 SPDSOCK_OBJS += spdsockddi.o spdsock.o spdsock_opt_data.o

575 IPSECESP_OBJS += ipsecespddi.o ipsecesp.o

577 IPSECAH_OBJS += ipsecahddi.o ipsecah.o sadb.o

579 SPPP_OBJS +=    sppp.o sppp_dlpi.o sppp_mod.o s_common.o

581 SPPPTUN_OBJS += sppptun.o sppptun_mod.o

583 SPPPASYN_OBJS += spppasyn.o spppasyn_mod.o

585 SPPPCOMP_OBJS += spppcomp.o spppcomp_mod.o deflate.o bsd-comp.o vjcompress.o \
586                 zlib.o

588 TCP_OBJS +=     tcpddi.o
```



```
590 TCP6_OBJS +=    tcp6ddi.o

592 NCA_OBJS +=     ncaddi.o

594 SDP_SOCK_MOD_OBJS += sockmod_sdp.o socksdp.o socksdpsubr.o

596 SCTP_SOCK_MOD_OBJS += sockmod_sctp.o socksctp.o socksctpsubr.o

598 PFP_SOCK_MOD_OBJS += sockmod_pfp.o

600 RDS_SOCK_MOD_OBJS += sockmod_rds.o

602 RDS_OBJS +=     rdsddi.o rdssubr.o rds_opt.o rds_ioctl.o

604 RDSIB_OBJS +=   rdsib.o rdsib_ib.o rdsib_cm.o rdsib_ep.o rdsib_buf.o \
605                 rdsib_debug.o rdsib_sc.o

607 RDSV3_OBJS +=   af_rds.o rdsv3_ddi.o bind.o loop.o threads.o connection.o \
608                 transport.o cong.o sysctl.o message.o rds_recv.o send.o \
609                 stats.o info.o page.o rdma_transport.o ib_ring.o ib_rdma.o \
610                 ib_recv.o ib.o ib_send.o ib_sysctl.o ib_stats.o ib_cm.o \
611                 rdsv3_sc.o rdsv3_debug.o rdsv3_impl.o rdma.o rdsv3_af_thr.o

613 ISER_OBJS +=    iser.o iser_cm.o iser_cq.o iser_ib.o iser_idm.o \
614                 iser_resource.o iser_xfer.o

616 UDP_OBJS +=     udpddi.o

618 UDP6_OBJS +=    udp6ddi.o

620 SY_OBJS +=      gentty.o

622 TCO_OBJS +=     ticots.o

624 TCOO_OBJS +=    ticotsord.o

626 TCL_OBJS +=     ticlts.o

628 TL_OBJS +=      tl.o

630 DUMP_OBJS +=    dump.o

632 BPF_OBJS +=     bpf.o bpf_filter.o bpf_mod.o bpf_dlt.o bpf_mac.o

634 CLONE_OBJS +=   clone.o

636 CN_OBJS +=      cons.o

638 DLD_OBJS +=     dld_drv.o dld_proto.o dld_str.o dld_flow.o

640 DLS_OBJS +=     dls.o dls_link.o dls_mod.o dls_stat.o dls_mgmt.o

642 GLD_OBJS +=     gld.o gldutil.o

644 MAC_OBJS +=     mac.o  mac_bcast.o mac_client.o mac_datapath_setup.o mac_flow.o
645                 mac_hio.o mac_mod.o mac_ndd.o mac_provider.o mac_sched.o \
646                 mac_protect.o mac_soft_ring.o mac_stat.o mac_util.o

648 MAC_6TO4_OBJS +=        mac_6to4.o

650 MAC_ETHER_OBJS +=       mac_ether.o

652 MAC_IPV4_OBJS +=        mac_ipv4.o

654 MAC_IPV6_OBJS +=        mac_ipv6.o
```

```
656 MAC_WIFI_OBJS +=        mac_wifi.o

658 MAC_IB_OBJS +=          mac_ib.o

660 IPTUN_OBJS +=   iptun_dev.o iptun_ctl.o iptun.o

662 AGGR_OBJS +=    aggr_dev.o aggr_ctl.o aggr_grp.o aggr_port.o \
663                 aggr_send.o aggr_recv.o aggr_lacp.o

665 SOFTMAC_OBJS += softmac_main.o softmac_ctl.o softmac_capab.o \
666                 softmac_dev.o softmac_stat.o softmac_pkt.o softmac_fp.o

668 NET80211_OBJS += net80211.o net80211_proto.o net80211_input.o \
669                  net80211_output.o net80211_node.o net80211_crypto.o \
670                  net80211_crypto_none.o net80211_crypto_wep.o net80211_ioctl.o \
671                  net80211_crypto_tkip.o net80211_crypto_ccmp.o  \
672                  net80211_ht.o

674 VNIC_OBJS +=    vnic_ctl.o vnic_dev.o

676 SIMNET_OBJS +=  simnet.o

678 IB_OBJS +=      ibnex.o ibnex_ioctl.o ibnex_hca.o

680 IBCM_OBJS +=    ibcm_impl.o ibcm_sm.o ibcm_ti.o ibcm_utils.o ibcm_path.o \
681                 ibcm_arp.o ibcm_arp_link.o

683 IBDM_OBJS +=    ibdm.o

685 IBDMA_OBJS +=   ibdma.o

687 IBMF_OBJS +=    ibmf.o ibmf_impl.o ibmf_dr.o ibmf_wqe.o ibmf_ud_dest.o ibmf_mod.
688                 ibmf_send.o ibmf_recv.o ibmf_handlers.o ibmf_trans.o \
689                 ibmf_timers.o ibmf_msg.o ibmf_utils.o ibmf_rmpp.o \
690                 ibmf_saa.o ibmf_saa_impl.o ibmf_saa_utils.o ibmf_saa_events.o

692 IBTL_OBJS +=    ibtl_impl.o ibtl_util.o ibtl_mem.o ibtl_handlers.o ibtl_qp.o \
693                 ibtl_cq.o ibtl_wr.o ibtl_hca.o ibtl_chan.o ibtl_cm.o \
694                 ibtl_mcg.o ibtl_ibnex.o ibtl_srq.o ibtl_part.o

696 TAVOR_OBJS +=   tavor.o tavor_agents.o tavor_cfg.o tavor_ci.o tavor_cmd.o \
697                 tavor_cq.o tavor_event.o tavor_ioctl.o tavor_misc.o \
698                 tavor_mr.o tavor_qp.o tavor_qpmod.o tavor_rsrc.o \
699                 tavor_srq.o tavor_stats.o tavor_umap.o tavor_wr.o

701 HERMON_OBJS +=  hermon.o hermon_agents.o hermon_cfg.o hermon_ci.o hermon_cmd.o \
702                 hermon_cq.o hermon_event.o hermon_ioctl.o hermon_misc.o \
703                 hermon_mr.o hermon_qp.o hermon_qpmod.o hermon_rsrc.o \
704                 hermon_srq.o hermon_stats.o hermon_umap.o hermon_wr.o \
705                 hermon_fcoib.o hermon_fm.o

707 DAPLT_OBJS +=   daplt.o

709 SOL_OFS_OBJS += sol_cma.o sol_ib_cma.o sol_uobj.o \
710                 sol_ofs_debug_util.o sol_ofs_gen_util.o \
711                 sol_kverbs.o

713 SOL_UCMA_OBJS +=        sol_ucma.o

715 SOL_UVERBS_OBJS +=      sol_uverbs.o sol_uverbs_comp.o sol_uverbs_event.o \
716                         sol_uverbs_hca.o sol_uverbs_qp.o

718 SOL_UMAD_OBJS += sol_umad.o

720 KSTAT_OBJS +=   kstat.o
```



```
722 KSYMS_OBJS +=   ksyms.o

724 INSTANCE_OBJS += inst_sync.o

726 IWSCN_OBJS +=   iwscons.o

728 LOFI_OBJS +=    lofi.o LzmaDec.o

730 FSSNAP_OBJS +=  fssnap.o

732 FSSNAPIF_OBJS += fssnap_if.o

734 MM_OBJS +=      mem.o

736 PHYSMEM_OBJS += physmem.o

738 OPTIONS_OBJS += options.o

740 WINLOCK_OBJS += winlockio.o

742 PM_OBJS +=      pm.o
743 SRN_OBJS +=     srn.o

745 PSEUDO_OBJS +=  pseudonex.o

747 RAMDISK_OBJS += ramdisk.o

749 LLC1_OBJS += llc1.o

751 USBKBM_OBJS += usbkbm.o

753 USBWCM_OBJS += usbwcm.o

755 BOFI_OBJS += bofi.o

757 HID_OBJS += hid.o

759 HWA_RC_OBJS += hwarc.o

761 USBSKEL_OBJS += usbskel.o

763 USBVC_OBJS += usbvc.o usbvc_v4l2.o

765 HIDPARSER_OBJS += hidparser.o

767 USB_AC_OBJS += usb_ac.o

769 USB_AS_OBJS += usb_as.o

771 USB_AH_OBJS += usb_ah.o

773 USBMS_OBJS += usbms.o

775 USBPRN_OBJS += usbprn.o

777 UGEN_OBJS += ugen.o

779 USBSER_OBJS += usbser.o usbser_rseq.o

781 USBSACM_OBJS += usbsacm.o

783 USBSER_KEYSPAN_OBJS += usbser_keyspan.o keyspan_dsd.o keyspan_pipe.o

785 USBS49_FW_OBJS += keyspan_49fw.o

787 USBSPRL_OBJS += usbser_pl2303.o pl2303_dsd.o
```

```
789 WUSB_CA_OBJS += wusb_ca.o

791 USBFTDI_OBJS += usbser_uftdi.o uftdi_dsd.o

793 USBECM_OBJS += usbecm.o

795 WC_OBJS += wscons.o vcons.o

797 VCONS_CONF_OBJS += vcons_conf.o

799 SCSI_OBJS +=    scsi_capabilities.o scsi_confsubr.o scsi_control.o \
800                 scsi_data.o scsi_fm.o scsi_hba.o scsi_reset_notify.o \
801                 scsi_resource.o scsi_subr.o scsi_transport.o scsi_watch.o \
802                 smp_transport.o

804 SCSI_VHCI_OBJS +=               scsi_vhci.o mpapi_impl.o scsi_vhci_tpgs.o

806 SCSI_VHCI_F_SYM_OBJS +=         sym.o

808 SCSI_VHCI_F_TPGS_OBJS +=        tpgs.o

810 SCSI_VHCI_F_ASYM_SUN_OBJS +=    asym_sun.o

812 SCSI_VHCI_F_SYM_HDS_OBJS +=     sym_hds.o

814 SCSI_VHCI_F_TAPE_OBJS +=        tape.o

816 SCSI_VHCI_F_TPGS_TAPE_OBJS +=   tpgs_tape.o

818 SGEN_OBJS +=    sgen.o

820 SMP_OBJS +=     smp.o

822 SATA_OBJS +=    sata.o

824 USBA_OBJS +=    hcdi.o  usba.o  usbai.o hubdi.o parser.o genconsole.o \
825                 usbai_pipe_mgmt.o usbai_req.o usbai_util.o usbai_register.o \
826                 usba_devdb.o usba10_calls.o usba_ugen.o whcdi.o wa.o
827 USBA_WITHOUT_WUSB_OBJS +=       hcdi.o  usba.o  usbai.o hubdi.o parser.o gencons
828                 usbai_pipe_mgmt.o usbai_req.o usbai_util.o usbai_register.o \
829                 usba_devdb.o usba10_calls.o usba_ugen.o

831 USBA10_OBJS +=  usba10.o

833 RSM_OBJS +=     rsm.o   rsmka_pathmanager.o     rsmka_util.o

835 RSMOPS_OBJS +=  rsmops.o

837 S1394_OBJS +=   t1394.o t1394_errmsg.o s1394.o s1394_addr.o s1394_asynch.o \
838                 s1394_bus_reset.o s1394_cmp.o s1394_csr.o s1394_dev_disc.o \
839                 s1394_fa.o s1394_fcp.o \
840                 s1394_hotplug.o s1394_isoch.o s1394_misc.o h1394.o nx1394.o

842 HCI1394_OBJS += hci1394.o hci1394_async.o hci1394_attach.o hci1394_buf.o \
843                 hci1394_csr.o hci1394_detach.o hci1394_extern.o \
844                 hci1394_ioctl.o hci1394_isoch.o hci1394_isr.o \
845                 hci1394_ixl_comp.o hci1394_ixl_isr.o hci1394_ixl_misc.o \
846                 hci1394_ixl_update.o hci1394_misc.o hci1394_ohci.o \
847                 hci1394_q.o hci1394_s1394if.o hci1394_tlabel.o \
848                 hci1394_tlist.o hci1394_vendor.o

850 AV1394_OBJS +=  av1394.o av1394_as.o av1394_async.o av1394_cfgrom.o \
851                 av1394_cmp.o av1394_fcp.o av1394_isoch.o av1394_isoch_chan.o \
852                 av1394_isoch_recv.o av1394_isoch_xmit.o av1394_list.o \
853                 av1394_queue.o
```



```
855 DCAM1394_OBJS += dcam.o dcam_frame.o dcam_param.o dcam_reg.o \
856                 dcam_ring_buff.o

858 SCSA1394_OBJS += hba.o sbp2_driver.o sbp2_bus.o

860 SBP2_OBJS += cfgrom.o sbp2.o

862 PMODEM_OBJS += pmodem.o pmodem_cis.o cis.o cis_callout.o cis_handlers.o cis_para

864 DSW_OBJS +=     dsw.o dsw_dev.o ii_tree.o

866 NCALL_OBJS +=   ncall.o \
867                 ncall_stub.o

869 RDC_OBJS +=     rdc.o \
870                 rdc_dev.o \
871                 rdc_io.o \
872                 rdc_clnt.o \
873                 rdc_prot_xdr.o \
874                 rdc_svc.o \
875                 rdc_bitmap.o \
876                 rdc_health.o \
877                 rdc_subr.o \
878                 rdc_diskq.o

880 RDCSRV_OBJS +=  rdcsrv.o

882 RDCSTUB_OBJS += rdc_stub.o

884 SDBC_OBJS +=    sd_bcache.o \
885                 sd_bio.o \
886                 sd_conf.o \
887                 sd_ft.o \
888                 sd_hash.o \
889                 sd_io.o \
890                 sd_misc.o \
891                 sd_pcu.o \
892                 sd_tdaemon.o \
893                 sd_trace.o \
894                 sd_iob_impl0.o \
895                 sd_iob_impl1.o \
896                 sd_iob_impl2.o \
897                 sd_iob_impl3.o \
898                 sd_iob_impl4.o \
899                 sd_iob_impl5.o \
900                 sd_iob_impl6.o \
901                 sd_iob_impl7.o \
902                 safestore.o \
903                 safestore_ram.o

905 NSCTL_OBJS +=   nsctl.o \
906                 nsc_cache.o \
907                 nsc_disk.o \
908                 nsc_dev.o \
909                 nsc_freeze.o \
910                 nsc_gen.o \
911                 nsc_mem.o \
912                 nsc_ncallio.o \
913                 nsc_power.o \
914                 nsc_resv.o \
915                 nsc_rmspin.o \
916                 nsc_solaris.o \
917                 nsc_trap.o \
918                 nsc_list.o
919 UNISTAT_OBJS += spuni.o \
```

```
 920                 spcs_s_k.o

 922 NSKERN_OBJS +=  nsc_ddi.o \
 923                 nsc_proc.o \
 924                 nsc_raw.o \
 925                 nsc_thread.o \
 926                 nskernd.o

 928 SV_OBJS +=      sv.o

 930 PMCS_OBJS += pmcs_attach.o pmcs_ds.o pmcs_intr.o pmcs_nvram.o pmcs_sata.o \
 931                 pmcs_scsa.o pmcs_smhba.o pmcs_subr.o pmcs_fwlog.o

 933 PMCS8001FW_C_OBJS +=    pmcs_fw_hdr.o
 934 PMCS8001FW_OBJS +=              $(PMCS8001FW_C_OBJS) SPCBoot.o ila.o firmware.o

 936 #
 937 #       Build up defines and paths.

 939 ST_OBJS +=      st.o    st_conf.o

 941 EMLXS_OBJS +=   emlxs_clock.o emlxs_dfc.o emlxs_dhchap.o emlxs_diag.o \
 942                 emlxs_download.o emlxs_dump.o emlxs_els.o emlxs_event.o \
 943                 emlxs_fcf.o emlxs_fcp.o emlxs_fct.o emlxs_hba.o emlxs_ip.o \
 944                 emlxs_mbox.o emlxs_mem.o emlxs_msg.o emlxs_node.o \
 945                 emlxs_pkt.o emlxs_sli3.o emlxs_sli4.o emlxs_solaris.o \
 946                 emlxs_thread.o

 948 EMLXS_FW_OBJS +=        emlxs_fw.o

 950 OCE_OBJS +=     oce_buf.o oce_fm.o oce_gld.o oce_hw.o oce_intr.o oce_main.o \
 951                 oce_mbx.o oce_mq.o oce_queue.o oce_rx.o oce_stat.o oce_tx.o \
 952                 oce_utils.o

 954 FCT_OBJS += discovery.o fct.o

 956 QLT_OBJS += 2400.o 2500.o 8100.o qlt.o qlt_dma.o

 958 SRPT_OBJS += srpt_mod.o srpt_ch.o srpt_cm.o srpt_ioc.o srpt_stp.o

 960 FCOE_OBJS += fcoe.o fcoe_eth.o fcoe_fc.o

 962 FCOET_OBJS += fcoet.o fcoet_eth.o fcoet_fc.o

 964 FCOEI_OBJS += fcoei.o fcoei_eth.o fcoei_lv.o

 966 ISCSIT_SHARED_OBJS += \
 967                 iscsit_common.o

 969 ISCSIT_OBJS +=  $(ISCSIT_SHARED_OBJS) \
 970                 iscsit.o iscsit_tgt.o iscsit_sess.o iscsit_login.o \
 971                 iscsit_text.o iscsit_isns.o iscsit_radiusauth.o \
 972                 iscsit_radiuspacket.o iscsit_auth.o iscsit_authclient.o

 974 PPPT_OBJS +=    alua_ic_if.o pppt.o pppt_msg.o pppt_tgt.o

 976 STMF_OBJS += lun_map.o stmf.o

 978 STMF_SBD_OBJS += sbd.o sbd_scsi.o sbd_pgr.o sbd_zvol.o

 980 SYSMSG_OBJS +=  sysmsg.o

 982 SES_OBJS +=     ses.o ses_sen.o ses_safte.o ses_ses.o

 984 TNF_OBJS +=     tnf_buf.o       tnf_trace.o     tnf_writer.o    trace_init.o \
 985                 trace_funcs.o   tnf_probe.o     tnf.o
```



```
 987 LOGINDMUX_OBJS += logindmux.o

 989 DEVINFO_OBJS += devinfo.o

 991 DEVPOLL_OBJS += devpoll.o

 993 DEVPOOL_OBJS += devpool.o

 995 I8042_OBJS +=   i8042.o

 997 KB8042_OBJS +=  \
 998                 at_keyprocess.o \
 999                 kb8042.o        \
1000                 kb8042_keytables.o

1002 MOUSE8042_OBJS += mouse8042.o

1004 FDC_OBJS +=     fdc.o

1006 ASY_OBJS +=     asy.o

1008 ECPP_OBJS +=    ecpp.o

1010 VUIDM3P_OBJS += vuidmice.o vuidm3p.o

1012 VUIDM4P_OBJS += vuidmice.o vuidm4p.o

1014 VUIDM5P_OBJS += vuidmice.o vuidm5p.o

1016 VUIDPS2_OBJS += vuidmice.o vuidps2.o

1018 HPCSVC_OBJS += hpcsvc.o

1020 PCIE_MISC_OBJS += pcie.o pcie_fault.o pcie_hp.o pciehpc.o pcishpc.o pcie_pwr.o p

1022 PCIHPNEXUS_OBJS += pcihp.o

1024 OPENEEPR_OBJS += openprom.o

1026 RANDOM_OBJS += random.o

1028 PSHOT_OBJS += pshot.o

1030 GEN_DRV_OBJS += gen_drv.o

1032 TCLIENT_OBJS += tclient.o

1034 TPHCI_OBJS += tphci.o

1036 TVHCI_OBJS += tvhci.o

1038 EMUL64_OBJS += emul64.o emul64_bsd.o

1040 FCP_OBJS += fcp.o

1042 FCIP_OBJS += fcip.o

1044 FCSM_OBJS += fcsm.o

1046 FCTL_OBJS += fctl.o

1048 FP_OBJS += fp.o

1050 QLC_OBJS += ql_api.o ql_debug.o ql_hba_fru.o ql_init.o ql_iocb.o ql_ioctl.o \
1051         ql_isr.o ql_mbx.o ql_nx.o ql_xioctl.o ql_fw_table.o
```

```
QLC_FW_2200_OBJS += ql_fw_2200.o

QLC_FW_2300_OBJS += ql_fw_2300.o

QLC_FW_2400_OBJS += ql_fw_2400.o

QLC_FW_2500_OBJS += ql_fw_2500.o

QLC_FW_6322_OBJS += ql_fw_6322.o

QLC_FW_8100_OBJS += ql_fw_8100.o

QLGE_OBJS += qlge.o qlge_dbg.o qlge_flash.o qlge_fm.o qlge_gld.o qlge_mpi.o

ZCONS_OBJS += zcons.o

NV_SATA_OBJS += nv_sata.o

SI3124_OBJS += si3124.o

AHCI_OBJS += ahci.o

PCIIDE_OBJS += pci-ide.o

PCEPP_OBJS += pcepp.o

CPC_OBJS += cpc.o

CPUID_OBJS += cpuid_drv.o

SYSEVENT_OBJS += sysevent.o

BL_OBJS += bl.o

DRM_OBJS += drm_sunmod.o drm_kstat.o drm_agpsupport.o \
            drm_auth.o drm_bufs.o drm_context.o drm_dma.o \
            drm_drawable.o drm_drv.o drm_fops.o drm_ioctl.o drm_irq.o \
            drm_lock.o drm_memory.o drm_msg.o drm_pci.o drm_scatter.o \
            drm_cache.o drm_gem.o drm_mm.o ati_pcigart.o

FM_OBJS += devfm.o devfm_machdep.o

RTLS_OBJS +=    rtls.o

#
#                       exec modules
#
AOUTEXEC_OBJS +=aout.o

ELFEXEC_OBJS += elf.o elf_notes.o old_notes.o

INTPEXEC_OBJS +=intp.o

SHBINEXEC_OBJS +=shbin.o

JAVAEXEC_OBJS +=java.o

#
#                       file system modules
#
AUTOFS_OBJS +=  auto_vfsops.o auto_vnops.o auto_subr.o auto_xdr.o auto_sys.o

CACHEFS_OBJS += cachefs_cnode.o         cachefs_cod.o \
                cachefs_dir.o           cachefs_dlog.o  cachefs_filegrp.o \
                cachefs_fscache.o       cachefs_ioctl.o cachefs_log.o \
```



```
                cachefs_module.o \
                cachefs_noopc.o         cachefs_resource.o \
                cachefs_strict.o \
                cachefs_subr.o          cachefs_vfsops.o \
                cachefs_vnops.o

DCFS_OBJS +=    dc_vnops.o

DEVFS_OBJS +=   devfs_subr.o    devfs_vfsops.o  devfs_vnops.o

DEV_OBJS  +=    sdev_subr.o     sdev_vfsops.o   sdev_vnops.o    \
                sdev_ptsops.o   sdev_zvolops.o  sdev_comm.o     \
                sdev_profile.o  sdev_ncache.o   sdev_netops.o   \
                sdev_ipnetops.o \
                sdev_vtops.o

CTFS_OBJS +=    ctfs_all.o ctfs_cdir.o ctfs_ctl.o ctfs_event.o \
                ctfs_latest.o ctfs_root.o ctfs_sym.o ctfs_tdir.o ctfs_tmpl.o

OBJFS_OBJS +=   objfs_vfs.o     objfs_root.o    objfs_common.o \
                objfs_odir.o    objfs_data.o

FDFS_OBJS +=    fdops.o

FIFO_OBJS +=    fifosubr.o      fifovnops.o

PIPE_OBJS +=    pipe.o

HSFS_OBJS +=    hsfs_node.o     hsfs_subr.o     hsfs_vfsops.o   hsfs_vnops.o \
                hsfs_susp.o     hsfs_rrip.o     hsfs_susp_subr.o

LOFS_OBJS +=    lofs_subr.o     lofs_vfsops.o   lofs_vnops.o

NAMEFS_OBJS +=  namevfs.o       namevno.o

NFS_OBJS +=     nfs_client.o    nfs_common.o    nfs_dump.o \
                nfs_subr.o      nfs_vfsops.o    nfs_vnops.o \
                nfs_xdr.o       nfs_sys.o       nfs_strerror.o \
                nfs3_vfsops.o   nfs3_vnops.o    nfs3_xdr.o \
                nfs_acl_vnops.o nfs_acl_xdr.o   nfs4_vfsops.o \
                nfs4_vnops.o    nfs4_xdr.o      nfs4_idmap.o \
                nfs4_shadow.o   nfs4_subr.o \
                nfs4_attr.o     nfs4_rnode.o    nfs4_client.o \
                nfs4_acache.o   nfs4_common.o   nfs4_client_state.o \
                nfs4_callback.o nfs4_recovery.o nfs4_client_secinfo.o \
                nfs4_client_debug.o     nfs_stats.o \
                nfs4_acl.o      nfs4_stub_vnops.o       nfs_cmd.o

NFSSRV_OBJS +=  nfs_server.o    nfs_srv.o       nfs3_srv.o \
                nfs_acl_srv.o   nfs_auth.o      nfs_auth_xdr.o \
                nfs_export.o    nfs_log.o       nfs_log_xdr.o \
                nfs4_srv.o      nfs4_state.o    nfs4_srv_attr.o \
                nfs4_srv_ns.o   nfs4_db.o       nfs4_srv_deleg.o \
                nfs4_deleg_ops.o nfs4_srv_readdir.o nfs4_dispatch.o

SMBSRV_SHARED_OBJS += \
                smb_inet.o \
                smb_match.o \
                smb_msgbuf.o \
                smb_oem.o \
                smb_string.o \
                smb_utf8.o \
                smb_door_legacy.o \
                smb_xdr.o \
                smb_token.o \
                smb_token_xdr.o \
```

```
1184                 smb_sid.o \
1185                 smb_native.o \
1186                 smb_netbios_util.o

1188 SMBSRV_OBJS +=  $(SMBSRV_SHARED_OBJS)                  \
1189                 smb_acl.o                              \
1190                 smb_alloc.o                            \
1191                 smb_close.o                            \
1192                 smb_common_open.o                      \
1193                 smb_common_transact.o                  \
1194                 smb_create.o                           \
1195                 smb_delete.o                           \
1196                 smb_directory.o                        \
1197                 smb_dispatch.o                         \
1198                 smb_echo.o                             \
1199                 smb_fem.o                              \
1200                 smb_find.o                             \
1201                 smb_flush.o                            \
1202                 smb_fsinfo.o                           \
1203                 smb_fsops.o                            \
1204                 smb_init.o                             \
1205                 smb_kdoor.o                            \
1206                 smb_kshare.o                           \
1207                 smb_kutil.o                            \
1208                 smb_lock.o                             \
1209                 smb_lock_byte_range.o                  \
1210                 smb_locking_andx.o                     \
1211                 smb_logoff_andx.o                      \
1212                 smb_mangle_name.o                      \
1213                 smb_mbuf_marshaling.o                  \
1214                 smb_mbuf_util.o                        \
1215                 smb_negotiate.o                        \
1216                 smb_net.o                              \
1217                 smb_node.o                             \
1218                 smb_nt_cancel.o                        \
1219                 smb_nt_create_andx.o                   \
1220                 smb_nt_transact_create.o               \
1221                 smb_nt_transact_ioctl.o                \
1222                 smb_nt_transact_notify_change.o        \
1223                 smb_nt_transact_quota.o                \
1224                 smb_nt_transact_security.o             \
1225                 smb_odir.o                             \
1226                 smb_ofile.o                            \
1227                 smb_open_andx.o                        \
1228                 smb_opipe.o                            \
1229                 smb_oplock.o                           \
1230                 smb_pathname.o                         \
1231                 smb_print.o                            \
1232                 smb_process_exit.o                     \
1233                 smb_query_fileinfo.o                   \
1234                 smb_read.o                             \
1235                 smb_rename.o                           \
1236                 smb_sd.o                               \
1237                 smb_seek.o                             \
1238                 smb_server.o                           \
1239                 smb_session.o                          \
1240                 smb_session_setup_andx.o               \
1241                 smb_set_fileinfo.o                     \
1242                 smb_signing.o                          \
1243                 smb_tree.o                             \
1244                 smb_trans2_create_directory.o          \
1245                 smb_trans2_dfs.o                       \
1246                 smb_trans2_find.o                      \
1247                 smb_tree_connect.o                     \
1248                 smb_unlock_byte_range.o                \
1249                 smb_user.o                             \
```



```
1250                 smb_vfs.o                              \
1251                 smb_vops.o                             \
1252                 smb_vss.o                              \
1253                 smb_write.o                            \
1254                 smb_write_raw.o

1256 PCFS_OBJS +=    pc_alloc.o      pc_dir.o        pc_node.o       pc_subr.o \
1257                 pc_vfsops.o     pc_vnops.o

1259 PROC_OBJS +=    prcontrol.o     prioctl.o       prsubr.o        prusrio.o \
1260                 prvfsops.o      prvnops.o

1262 MNTFS_OBJS +=   mntvfsops.o     mntvnops.o

1264 SHAREFS_OBJS += sharetab.o      sharefs_vfsops.o        sharefs_vnops.o

1266 SPEC_OBJS +=    specsubr.o      specvfsops.o    specvnops.o

1268 SOCK_OBJS +=    socksubr.o      sockvfsops.o    sockparams.o    \
1269                 socksyscalls.o  socktpi.o       sockstr.o \
1270                 sockcommon_vnops.o      sockcommon_subr.o \
1271                 sockcommon_sops.o       sockcommon.o    \
1272                 sock_notsupp.o  socknotify.o \
1273                 nl7c.o          nl7curi.o       nl7chttp.o      nl7clogd.o \
1274                 nl7cnca.o       sodirect.o      sockfilter.o

1276 TMPFS_OBJS +=   tmp_dir.o       tmp_subr.o      tmp_tnode.o     tmp_vfsops.o \
1277                 tmp_vnops.o

1279 UDFS_OBJS +=    udf_alloc.o     udf_bmap.o      udf_dir.o       \
1280                 udf_inode.o     udf_subr.o      udf_vfsops.o    \
1281                 udf_vnops.o

1283 UFS_OBJS +=     ufs_alloc.o     ufs_bmap.o      ufs_dir.o       ufs_xattr.o \
1284                 ufs_inode.o     ufs_subr.o      ufs_tables.o    ufs_vfsops.o \
1285                 ufs_vnops.o     quota.o         quotacalls.o    quota_ufs.o \
1286                 ufs_filio.o     ufs_lockfs.o    ufs_thread.o    ufs_trans.o \
1287                 ufs_acl.o       ufs_panic.o     ufs_directio.o  ufs_log.o \
1288                 ufs_extvnops.o  ufs_snap.o      lufs.o          lufs_thread.o \
1289                 lufs_log.o      lufs_map.o      lufs_top.o      lufs_debug.o
1290 VSCAN_OBJS +=   vscan_drv.o     vscan_svc.o vscan_door.o

1292 NSMB_OBJS +=    smb_conn.o      smb_dev.o       smb_iod.o       smb_pass.o \
1293                 smb_rq.o        smb_sign.o      smb_smb.o       smb_subrs.o \
1294                 smb_time.o      smb_tran.o      smb_trantcp.o   smb_usr.o \
1295                 subr_mchain.o

1297 SMBFS_COMMON_OBJS += smbfs_ntacl.o
1298 SMBFS_OBJS +=    smbfs_vfsops.o  smbfs_vnops.o   smbfs_node.o    \
1299                 smbfs_acl.o     smbfs_client.o  smbfs_smb.o     \
1300                 smbfs_subr.o    smbfs_subr2.o   \
1301                 smbfs_rwlock.o  smbfs_xattr.o   \
1302                 $(SMBFS_COMMON_OBJS)


1305 #
1306 #                       LVM modules
1307 #
1308 MD_OBJS += md.o md_error.o md_ioctl.o md_mddb.o md_names.o \
1309         md_med.o md_rename.o md_subr.o

1311 MD_COMMON_OBJS = md_convert.o md_crc.o md_revchk.o

1313 MD_DERIVED_OBJS = metamed_xdr.o meta_basic_xdr.o

1315 SOFTPART_OBJS += sp.o sp_ioctl.o
```

```
1317 STRIPE_OBJS += stripe.o stripe_ioctl.o

1319 HOTSPARES_OBJS += hotspares.o

1321 RAID_OBJS += raid.o raid_ioctl.o raid_replay.o raid_resync.o raid_hotspare.o

1323 MIRROR_OBJS += mirror.o mirror_ioctl.o mirror_resync.o

1325 NOTIFY_OBJS += md_notify.o

1327 TRANS_OBJS += mdtrans.o trans_ioctl.o trans_log.o

1329 ZFS_COMMON_OBJS +=             \
1330         arc.o                  \
1331         bplist.o               \
1332         bpobj.o                \
1333         bptree.o               \
1334         dbuf.o                 \
1335         ddt.o                  \
1336         ddt_zap.o              \
1337         dmu.o                  \
1338         dmu_diff.o             \
1339         dmu_send.o             \
1340         dmu_object.o           \
1341         dmu_objset.o           \
1342         dmu_traverse.o         \
1343         dmu_tx.o               \
1344         dnode.o                \
1345         dnode_sync.o           \
1346         dsl_dir.o              \
1347         dsl_dataset.o          \
1348         dsl_deadlist.o         \
1349         dsl_destroy.o          \
1350         dsl_pool.o             \
1351         dsl_synctask.o         \
1352         dsl_userhold.o         \
1353         dmu_zfetch.o           \
1354         dsl_deleg.o            \
1355         dsl_prop.o             \
1356         dsl_scan.o             \
1357         zfeature.o             \
1358         gzip.o                 \
1359         lz4.o                  \
1360         lzjb.o                 \
1361         metaslab.o             \
1362         refcount.o             \
1363         rrwlock.o              \
1364         sa.o                   \
1365         sha256.o               \
1366         spa.o                  \
1367         spa_config.o           \
1368         spa_errlog.o           \
1369         spa_history.o          \
1370         spa_misc.o             \
1371         space_map.o            \
1372         txg.o                  \
1373         uberblock.o            \
1374         unique.o               \
1375         vdev.o                 \
1376         vdev_cache.o           \
1377         vdev_file.o            \
1378         vdev_label.o           \
1379         vdev_mirror.o          \
1380         vdev_missing.o         \
1381         vdev_queue.o           \
```



```
1382         vdev_raidz.o           \
1383         vdev_root.o            \
1384         zap.o                  \
1385         zap_leaf.o             \
1386         zap_micro.o            \
1387         zfs_byteswap.o         \
1388         zfs_debug.o            \
1389         zfs_fm.o               \
1390         zfs_fuid.o             \
1391         zfs_sa.o               \
1392         zfs_znode.o            \
1393         zil.o                  \
1394         zio.o                  \
1395         zio_checksum.o         \
1396         zio_compress.o         \
1397         zio_inject.o           \
1398         zle.o                  \
1399         zrlock.o

1401 ZFS_SHARED_OBJS +=             \
1402         zfeature_common.o      \
1403         zfs_comutil.o          \
1404         zfs_deleg.o            \
1405         zfs_fletcher.o         \
1406         zfs_namecheck.o        \
1407         zfs_prop.o             \
1408         zpool_prop.o           \
1409         zprop_common.o

1411 ZFS_OBJS +=                    \
1412         $(ZFS_COMMON_OBJS)     \
1413         $(ZFS_SHARED_OBJS)     \
1414         vdev_disk.o            \
1415         zfs_acl.o              \
1416         zfs_ctldir.o           \
1417         zfs_dir.o              \
1418         zfs_ioctl.o            \
1419         zfs_log.o              \
1420         zfs_onexit.o           \
1421         zfs_replay.o           \
1422         zfs_rlock.o            \
1423         zfs_vfsops.o           \
1424         zfs_vnops.o            \
1425         zvol.o

1427 ZUT_OBJS +=                    \
1428         zut.o

1430 #
1431 #                       streams modules
1432 #
1433 BUFMOD_OBJS     +=      bufmod.o

1435 CONNLD_OBJS +=  connld.o

1437 DEDUMP_OBJS +=  dedump.o

1439 DRCOMPAT_OBJS +=        drcompat.o

1441 LDLINUX_OBJS += ldlinux.o

1443 LDTERM_OBJS +=  ldterm.o uwidth.o

1445 PCKT_OBJS +=    pckt.o

1447 PFMOD_OBJS +=   pfmod.o
```

```
1449 PTEM_OBJS +=    ptem.o

1451 REDIRMOD_OBJS += strredirm.o

1453 TIMOD_OBJS +=   timod.o

1455 TIRDWR_OBJS +=  tirdwr.o

1457 TTCOMPAT_OBJS +=ttcompat.o

1459 LOG_OBJS +=     log.o

1461 PIPEMOD_OBJS += pipemod.o

1463 RPCMOD_OBJS +=  rpcmod.o        clnt_cots.o     clnt_clts.o \
1464                 clnt_gen.o      clnt_perr.o     mt_rpcinit.o    rpc_calmsg.o \
1465                 rpc_prot.o      rpc_sztypes.o   rpc_subr.o      rpcb_prot.o \
1466                 svc.o           svc_clts.o      svc_gen.o       svc_cots.o \
1467                 rpcsys.o        xdr_sizeof.o    clnt_rdma.o     svc_rdma.o \
1468                 xdr_rdma.o      rdma_subr.o     xdrrdma_sizeof.o

1470 TLIMOD_OBJS +=  tlimod.o        t_kalloc.o      t_kbind.o       t_kclose.o \
1471                 t_kconnect.o    t_kfree.o       t_kgtstate.o    t_kopen.o \
1472                 t_krcvudat.o    t_ksndudat.o    t_kspoll.o      t_kunbind.o \
1473                 t_kutil.o

1475 RLMOD_OBJS += rlmod.o

1477 TELMOD_OBJS += telmod.o

1479 CRYPTMOD_OBJS += cryptmod.o

1481 KB_OBJS +=      kbd.o           keytables.o

1483 #
1484 #                       ID mapping module
1485 #
1486 IDMAP_OBJS +=   idmap_mod.o     idmap_kapi.o    idmap_xdr.o     idmap_cache.o

1488 #
1489 #                       scheduling class modules
1490 #
1491 SDC_OBJS +=             sysdc.o

1493 RT_OBJS +=              rt.o
1494 RT_DPTBL_OBJS +=        rt_dptbl.o

1496 TS_OBJS +=              ts.o
1497 TS_DPTBL_OBJS +=        ts_dptbl.o

1499 IA_OBJS +=              ia.o

1501 FSS_OBJS +=             fss.o

1503 FX_OBJS +=              fx.o
1504 FX_DPTBL_OBJS +=        fx_dptbl.o

1506 #
1507 #                       Inter-Process Communication (IPC) modules
1508 #
1509 IPC_OBJS +=     ipc.o

1511 IPCMSG_OBJS +=  msg.o

1513 IPCSEM_OBJS +=  sem.o
```



```
1515 IPCSHM_OBJS +=  shm.o

1517 #
1518 #                       bignum module
1519 #
1520 COMMON_BIGNUM_OBJS += bignum_mod.o bignumimpl.o

1522 BIGNUM_OBJS += $(COMMON_BIGNUM_OBJS) $(BIGNUM_PSR_OBJS)

1524 #
1525 #                       kernel cryptographic framework
1526 #
1527 KCF_OBJS +=     kcf.o kcf_callprov.o kcf_cbufcall.o kcf_cipher.o kcf_crypto.o \
1528                 kcf_cryptoadm.o kcf_ctxops.o kcf_digest.o kcf_dual.o \
1529                 kcf_keys.o kcf_mac.o kcf_mech_tabs.o kcf_miscapi.o \
1530                 kcf_object.o kcf_policy.o kcf_prov_lib.o kcf_prov_tabs.o \
1531                 kcf_sched.o kcf_session.o kcf_sign.o kcf_spi.o kcf_verify.o \
1532                 kcf_random.o modes.o ecb.o cbc.o ctr.o ccm.o gcm.o \
1533                 fips_random.o

1535 CRYPTOADM_OBJS += cryptoadm.o

1537 CRYPTO_OBJS +=  crypto.o

1539 DPROV_OBJS +=   dprov.o

1541 DCA_OBJS +=     dca.o dca_3des.o dca_debug.o dca_dsa.o dca_kstat.o dca_rng.o \
1542                 dca_rsa.o

1544 AESPROV_OBJS += aes.o aes_impl.o aes_modes.o

1546 ARCFOURPROV_OBJS += arcfour.o arcfour_crypt.o

1548 BLOWFISHPROV_OBJS += blowfish.o blowfish_impl.o

1550 ECCPROV_OBJS += ecc.o ec.o ec2_163.o ec2_mont.o ecdecode.o ecl_mult.o \
1551                 ecp_384.o ecp_jac.o ec2_193.o ecl.o ecp_192.o ecp_521.o \
1552                 ecp_jm.o ec2_233.o ecl_curve.o ecp_224.o ecp_aff.o \
1553                 ecp_mont.o ec2_aff.o ec_naf.o ecl_gf.o ecp_256.o mp_gf2m.o \
1554                 mpi.o mplogic.o mpmontg.o mpprime.o oid.o \
1555                 secitem.o ec2_test.o ecp_test.o

1557 RSAPROV_OBJS += rsa.o rsa_impl.o pkcs1.o

1559 SWRANDPROV_OBJS += swrand.o

1561 #
1562 #                       kernel SSL
1563 #
1564 KSSL_OBJS +=    kssl.o kssl_ioctl.o

1566 KSSL_SOCKFIL_MOD_OBJS += ksslfilter.o ksslapi.o ksslrec.o

1568 #
1569 #                       misc. modules
1570 #

1572 C2AUDIT_OBJS += adr.o audit.o audit_event.o audit_io.o \
1573                 audit_path.o audit_start.o audit_syscalls.o audit_token.o \
1574                 audit_mem.o

1576 PCIC_OBJS +=    pcic.o

1578 RPCSEC_OBJS +=  secmod.o        sec_clnt.o      sec_svc.o       sec_gen.o \
1579                 auth_des.o      auth_kern.o     auth_none.o     auth_loopb.o\
```

```
1580                 authdesprt.o    authdesubr.o    authu_prot.o \
1581                 key_call.o      key_prot.o      svc_authu.o     svcauthdes.o

1583 RPCSEC_GSS_OBJS +=      rpcsec_gssmod.o rpcsec_gss.o rpcsec_gss_misc.o \
1584                 rpcsec_gss_utils.o svc_rpcsec_gss.o

1586 CONSCONFIG_OBJS += consconfig.o

1588 CONSCONFIG_DACF_OBJS  += consconfig_dacf.o consplat.o

1590 TEM_OBJS += tem.o tem_safe.o 6x10.o 7x14.o 12x22.o

1592 KBTRANS_OBJS +=                          \
1593                 kbtrans.o               \
1594                 kbtrans_keytables.o     \
1595                 kbtrans_polled.o        \
1596                 kbtrans_streams.o       \
1597                 usb_keytables.o

1599 KGSSD_OBJS +=   gssd_clnt_stubs.o gssd_handle.o gssd_prot.o \
1600                 gss_display_name.o gss_release_name.o gss_import_name.o \
1601                 gss_release_buffer.o gss_release_oid_set.o gen_oids.o gssdmod.o

1603 KGSSD_DERIVED_OBJS = gssd_xdr.o

1605 KGSS_DUMMY_OBJS += dmech.o

1607 KSOCKET_OBJS += ksocket.o ksocket_mod.o

1609 CRYPTO= cksumtypes.o decrypt.o encrypt.o encrypt_length.o etypes.o \
1610         nfold.o verify_checksum.o prng.o block_size.o make_checksum.o\
1611         checksum_length.o hmac.o default_state.o mandatory_sumtype.o

1613 # crypto/des
1614 CRYPTO_DES= f_cbc.o f_cksum.o f_parity.o weak_key.o d3_cbc.o ef_crypto.o

1616 CRYPTO_DK= checksum.o derive.o dk_decrypt.o dk_encrypt.o

1618 CRYPTO_ARCFOUR= k5_arcfour.o

1620 # crypto/enc_provider
1621 CRYPTO_ENC= des.o des3.o arcfour_provider.o aes_provider.o

1623 # crypto/hash_provider
1624 CRYPTO_HASH= hash_kef_generic.o hash_kmd5.o hash_crc32.o hash_ksha1.o

1626 # crypto/keyhash_provider
1627 CRYPTO_KEYHASH= descbc.o k5_kmd5des.o k_hmac_md5.o

1629 # crypto/crc32
1630 CRYPTO_CRC32= crc32.o

1632 # crypto/old
1633 CRYPTO_OLD= old_decrypt.o old_encrypt.o

1635 # crypto/raw
1636 CRYPTO_RAW= raw_decrypt.o raw_encrypt.o

1638 K5_KRB= kfree.o copy_key.o \
1639         parse.o init_ctx.o \
1640         ser_adata.o ser_addr.o \
1641         ser_auth.o ser_cksum.o \
1642         ser_key.o ser_princ.o \
1643         serialize.o unparse.o \
1644         ser_actx.o
```



```
1646 K5_OS=  timeofday.o toffset.o \
1647         init_os_ctx.o c_ustime.o

1649 SEAL=
1650 # EXPORT DELETE START
1649 SEAL=   seal.o unseal.o
1652 # EXPORT DELETE END

1651 MECH=   delete_sec_context.o \
1652         import_sec_context.o \
1653         gssapi_krb5.o \
1654         k5seal.o k5unseal.o k5sealv3.o \
1655         ser_sctx.o \
1656         sign.o \
1657         util_crypt.o  \
1658         util_validate.o  util_ordering.o  \
1659         util_seqnum.o util_set.o util_seed.o \
1660         wrap_size_limit.o verify.o



1664 MECH_GEN= util_token.o


1667 KGSS_KRB5_OBJS += krb5mech.o \
1668         $(MECH) $(SEAL) $(MECH_GEN) \
1669         $(CRYPTO) $(CRYPTO_DES) $(CRYPTO_DK) $(CRYPTO_ARCFOUR) \
1670         $(CRYPTO_ENC) $(CRYPTO_HASH) \
1671         $(CRYPTO_KEYHASH) $(CRYPTO_CRC32) \
1672         $(CRYPTO_OLD) \
1673         $(CRYPTO_RAW) $(K5_KRB) $(K5_OS)

1675 DES_OBJS +=     des_crypt.o des_impl.o des_ks.o des_soft.o

1677 DLBOOT_OBJS +=  bootparam_xdr.o nfs_dlinet.o scan.o

1679 KRTLD_OBJS +=   kobj_bootflags.o getoptstr.o \
1680                 kobj.o kobj_kdi.o kobj_lm.o kobj_subr.o

1682 MOD_OBJS +=     modctl.o modsubr.o modsysfile.o modconf.o modhash.o

1684 STRPLUMB_OBJS += strplumb.o

1686 CPR_OBJS +=     cpr_driver.o cpr_dump.o \
1687                 cpr_main.o cpr_misc.o cpr_mod.o cpr_stat.o \
1688                 cpr_uthread.o

1690 PROF_OBJS +=    prf.o

1692 SE_OBJS += se_driver.o

1694 SYSACCT_OBJS += acct.o

1696 ACCTCTL_OBJS += acctctl.o

1698 EXACCTSYS_OBJS += exacctsys.o

1700 KAIO_OBJS += aio.o

1702 PCMCIA_OBJS += pcmcia.o cs.o cis.o cis_callout.o cis_handlers.o cis_params.o

1704 BUSRA_OBJS += busra.o

1706 PCS_OBJS += pcs.o

1708 PCAN_OBJS += pcan.o
```

```
1710 PCATA_OBJS += pcide.o pcdisk.o pclabel.o pcata.o

1712 PCSER_OBJS += pcser.o pcser_cis.o

1714 PCWL_OBJS += pcwl.o

1716 PSET_OBJS +=    pset.o

1718 OHCI_OBJS += ohci.o ohci_hub.o ohci_polled.o

1720 UHCI_OBJS += uhci.o uhciutil.o uhcitgt.o uhcihub.o uhcipolled.o

1722 EHCI_OBJS += ehci.o ehci_hub.o ehci_xfer.o ehci_intr.o ehci_util.o ehci_polled.o

1724 HUBD_OBJS += hubd.o

1726 USB_MID_OBJS += usb_mid.o

1728 USB_IA_OBJS += usb_ia.o

1730 UWBA_OBJS += uwba.o uwbai.o

1732 SCSA2USB_OBJS += scsa2usb.o usb_ms_bulkonly.o usb_ms_cbi.o

1734 HWAHC_OBJS += hwahc.o hwahc_util.o

1736 WUSB_DF_OBJS += wusb_df.o
1737 WUSB_FWMOD_OBJS += wusb_fwmod.o

1739 IPF_OBJS += ip_fil_solaris.o fil.o solaris.o ip_state.o ip_frag.o ip_nat.o \
1740             ip_proxy.o ip_auth.o ip_pool.o ip_htable.o ip_lookup.o \
1741             ip_log.o misc.o ip_compat.o ip_nat6.o drand48.o

1743 IBD_OBJS +=     ibd.o ibd_cm.o

1745 EIBNX_OBJS +=   enx_main.o enx_hdlrs.o enx_ibt.o enx_log.o enx_fip.o \
1746                 enx_misc.o enx_q.o enx_ctl.o

1748 EOIB_OBJS +=    eib_adm.o eib_chan.o eib_cmn.o eib_ctl.o eib_data.o \
1749                 eib_fip.o eib_ibt.o eib_log.o eib_mac.o eib_main.o \
1750                 eib_rsrc.o eib_svc.o eib_vnic.o

1752 DLPISTUB_OBJS += dlpistub.o

1754 SDP_OBJS +=     sdpddi.o

1756 TRILL_OBJS +=   trill.o

1758 CTF_OBJS += ctf_create.o ctf_decl.o ctf_error.o ctf_hash.o ctf_labels.o \
1759         ctf_lookup.o ctf_open.o ctf_types.o ctf_util.o ctf_subr.o ctf_mod.o

1761 SMBIOS_OBJS += smb_error.o smb_info.o smb_open.o smb_subr.o smb_dev.o

1763 RPCIB_OBJS += rpcib.o

1765 KMDB_OBJS += kdrv.o

1767 AFE_OBJS += afe.o

1769 BGE_OBJS += bge_main2.o bge_chip2.o bge_kstats.o bge_log.o bge_ndd.o \
1770                 bge_atomic.o bge_mii.o bge_send.o bge_recv2.o bge_mii_5906.o

1772 DMFE_OBJS += dmfe_log.o dmfe_main.o dmfe_mii.o

1774 EFE_OBJS += efe.o
```



```
1776 ELXL_OBJS += elxl.o

1778 HME_OBJS += hme.o

1780 IXGB_OBJS += ixgb.o ixgb_atomic.o ixgb_chip.o ixgb_gld.o ixgb_kstats.o \
1781                 ixgb_log.o ixgb_ndd.o ixgb_rx.o ixgb_tx.o ixgb_xmii.o

1783 NGE_OBJS += nge_main.o nge_atomic.o nge_chip.o nge_ndd.o nge_kstats.o \
1784                 nge_log.o nge_rx.o nge_tx.o nge_xmii.o

1786 PCN_OBJS += pcn.o

1788 RGE_OBJS += rge_main.o rge_chip.o rge_ndd.o rge_kstats.o rge_log.o rge_rxtx.o

1790 URTW_OBJS += urtw.o

1792 ARN_OBJS += arn_hw.o arn_eeprom.o arn_mac.o arn_calib.o arn_ani.o arn_phy.o arn_
1793                 arn_main.o arn_recv.o arn_xmit.o arn_rc.o

1795 ATH_OBJS += ath_aux.o ath_main.o ath_osdep.o ath_rate.o

1797 ATU_OBJS += atu.o

1799 IPW_OBJS += ipw2100_hw.o ipw2100.o

1801 IWI_OBJS += ipw2200_hw.o ipw2200.o

1803 IWH_OBJS += iwh.o

1805 IWK_OBJS += iwk2.o

1807 IWP_OBJS += iwp.o

1809 MWL_OBJS += mwl.o

1811 MWLFW_OBJS += mwlfw_mode.o

1813 WPI_OBJS += wpi.o

1815 RAL_OBJS += rt2560.o ral_rate.o

1817 RUM_OBJS += rum.o

1819 RWD_OBJS += rt2661.o

1821 RWN_OBJS += rt2860.o

1823 UATH_OBJS += uath.o

1825 UATHFW_OBJS += uathfw_mod.o

1827 URAL_OBJS += ural.o

1829 RTW_OBJS += rtw.o smc93cx6.o rtwphy.o rtwphyio.o

1831 ZYD_OBJS += zyd.o zyd_usb.o zyd_hw.o zyd_fw.o

1833 MXFE_OBJS += mxfe.o

1835 MPTSAS_OBJS += mptsas.o mptsas_impl.o mptsas_init.o mptsas_raid.o mptsas_smhba.o

1837 SFE_OBJS += sfe.o sfe_util.o

1839 BFE_OBJS += bfe.o
```

```
1841 BRIDGE_OBJS += bridge.o

1843 IDM_SHARED_OBJS += base64.o

1845 IDM_OBJS +=     $(IDM_SHARED_OBJS) \
1846                 idm.o idm_impl.o idm_text.o idm_conn_sm.o idm_so.o

1848 VR_OBJS += vr.o

1850 ATGE_OBJS += atge_main.o atge_l1e.o atge_mii.o atge_l1.o atge_l1c.o

1852 YGE_OBJS = yge.o

1854 #
1855 #       Build up defines and paths.
1856 #
1857 LINT_DEFS       += -Dunix

1859 #
1860 #       This duality can be removed when the native and target compilers
1861 #       are the same (or at least recognize the same command line syntax!)
1862 #       It is a bug in the current compilation system that the assember
1863 #       can't process the -Y I, flag.
1864 #
1865 NATIVE_INC_PATH += $(INC_PATH) $(CCYFLAG)$(UTSBASE)/common
1866 AS_INC_PATH     += $(INC_PATH) -I$(UTSBASE)/common
1867 INCLUDE_PATH    += $(INC_PATH) $(CCYFLAG)$(UTSBASE)/common

1869 PCIEB_OBJS += pcieb.o

1871 #       Chelsio N110 10G NIC driver module
1872 #
1873 CH_OBJS = ch.o glue.o pe.o sge.o

1875 CH_COM_OBJS =   ch_mac.o ch_subr.o cspi.o espi.o ixf1010.o mc3.o mc4.o mc5.o \
1876                 mv88e1xxx.o mv88x201x.o my3126.o pm3393.o tp.o ulp.o \
1877                 vsc7321.o vsc7326.o xpak.o

1879 #
1880 #       Chelsio Terminator 4 10G NIC nexus driver module
1881 #
1882 CXGBE_FW_OBJS  =        t4_fw.o t4_cfg.o
1883 CXGBE_COM_OBJS =        t4_hw.o common.o
1884 CXGBE_NEX_OBJS =        t4_nexus.o t4_sge.o t4_mac.o t4_ioctl.o shared.o \
1885                         t4_l2t.o adapter.o osdep.o

1887 #
1888 #       Chelsio Terminator 4 10G NIC driver module
1889 #
1890 CXGBE_OBJS =    cxgbe.o

1892 #
1893 #       PCI strings file
1894 #
1895 PCI_STRING_OBJS = pci_strings.o

1897 NET_DACF_OBJS += net_dacf.o

1899 #
1900 #       Xframe 10G NIC driver module
1901 #
1902 XGE_OBJS = xge.o xgell.o

1904 XGE_HAL_OBJS =  xgehal-channel.o xgehal-fifo.o xgehal-ring.o  xgehal-config.o \
1905                 xgehal-driver.o  xgehal-mm.o xgehal-stats.o  xgehal-device.o \
1906                 xge-queue.o  xgehal-mgmt.o xgehal-mgmtaux.o
```



```
1908 #
1909 #       e1000g module
1910 #
1911 E1000G_OBJS +=  e1000_80003es2lan.o e1000_82540.o e1000_82541.o e1000_82542.o \
1912                 e1000_82543.o e1000_82571.o e1000_api.o e1000_ich8lan.o \
1913                 e1000_mac.o e1000_manage.o e1000_nvm.o e1000_osdep.o \
1914                 e1000_phy.o e1000g_debug.o e1000g_main.o e1000g_alloc.o \
1915                 e1000g_tx.o e1000g_rx.o e1000g_stat.o

1917 #
1918 #       Intel 82575 1G NIC driver module
1919 #
1920 IGB_OBJS =      igb_82575.o igb_api.o igb_mac.o igb_manage.o \
1921                 igb_nvm.o igb_osdep.o igb_phy.o igb_buf.o \
1922                 igb_debug.o igb_gld.o igb_log.o igb_main.o \
1923                 igb_rx.o igb_stat.o igb_tx.o

1925 #
1926 #       Intel Pro/100 NIC driver module
1927 #
1928 IPRB_OBJS =     iprb.o

1930 #
1931 #       Intel 10GbE PCIE NIC driver module
1932 #
1933 IXGBE_OBJS =    ixgbe_82598.o ixgbe_82599.o ixgbe_api.o         \
1934                 ixgbe_common.o ixgbe_phy.o                      \
1935                 ixgbe_buf.o ixgbe_debug.o ixgbe_gld.o           \
1936                 ixgbe_log.o ixgbe_main.o                        \
1937                 ixgbe_osdep.o ixgbe_rx.o ixgbe_stat.o           \
1938                 ixgbe_tx.o  ixgbe_x540.o ixgbe_mbx.o

1940 #
1941 #       NIU 10G/1G driver module
1942 #
1943 NXGE_OBJS =     nxge_mac.o nxge_ipp.o nxge_rxdma.o               \
1944                 nxge_txdma.o nxge_txc.o nxge_main.o             \
1945                 nxge_hw.o nxge_fzc.o nxge_virtual.o             \
1946                 nxge_send.o nxge_classify.o nxge_fflp.o         \
1947                 nxge_fflp_hash.o nxge_ndd.o nxge_kstats.o       \
1948                 nxge_zcp.o nxge_fm.o nxge_espc.o nxge_hv.o      \
1949                 nxge_hio.o nxge_hio_guest.o nxge_intr.o

1951 NXGE_NPI_OBJS = \
1952                 npi.o npi_mac.o npi_ipp.o                       \
1953                 npi_txdma.o npi_rxdma.o npi_txc.o               \
1954                 npi_zcp.o npi_espc.o npi_fflp.o                 \
1955                 npi_vir.o

1957 NXGE_HCALL_OBJS =       \
1958                 nxge_hcall.o

1960 #
1961 # Virtio modules
1962 #

1964 # Virtio core
1965 VIRTIO_OBJS = virtio.o

1967 # Virtio block driver
1968 VIOBLK_OBJS = vioblk.o

1970 #
1971 #       kiconv modules
1972 #
```

```
1973 KICONV_EMEA_OBJS += kiconv_emea.o

1975 KICONV_JA_OBJS += kiconv_ja.o

1977 KICONV_KO_OBJS += kiconv_cck_common.o kiconv_ko.o

1979 KICONV_SC_OBJS += kiconv_cck_common.o kiconv_sc.o

1981 KICONV_TC_OBJS += kiconv_cck_common.o kiconv_tc.o

1983 #
1984 #       AAC module
1985 #
1986 AAC_OBJS = aac.o aac_ioctl.o

1988 #
1989 #       sdcard modules
1990 #
1991 SDA_OBJS =      sda_cmd.o sda_host.o sda_init.o sda_mem.o sda_mod.o sda_slot.o
1992 SDHOST_OBJS =   sdhost.o

1994 #
1995 #       hxge 10G driver module
1996 #
1997 HXGE_OBJS =     hxge_main.o hxge_vmac.o hxge_send.o              \
1998                 hxge_txdma.o hxge_rxdma.o hxge_virtual.o        \
1999                 hxge_fm.o hxge_fzc.o hxge_hw.o hxge_kstats.o    \
2000                 hxge_ndd.o hxge_pfc.o                           \
2001                 hpi.o hpi_vmac.o hpi_rxdma.o hpi_txdma.o        \
2002                 hpi_vir.o hpi_pfc.o

2004 #
2005 #       MEGARAID_SAS module
2006 #
2007 MEGA_SAS_OBJS = megaraid_sas.o

2009 #
2010 #       MR_SAS module
2011 #
2012 MR_SAS_OBJS = ld_pd_map.o mr_sas.o mr_sas_tbolt.o mr_sas_list.o

2014 #
2015 #       ISCSI_INITIATOR module
2016 #
2017 ISCSI_INITIATOR_OBJS =  chap.o iscsi_io.o iscsi_thread.o        \
2018                         iscsi_ioctl.o iscsid.o iscsi.o          \
2019                         iscsi_login.o isns_client.o iscsiAuthClient.o   \
2020                         iscsi_lun.o iscsiAuthClientGlue.o       \
2021                         iscsi_net.o nvfile.o iscsi_cmd.o        \
2022                         iscsi_queue.o persistent.o iscsi_conn.o \
2023                         iscsi_sess.o radius_auth.o iscsi_crc.o  \
2024                         iscsi_stats.o radius_packet.o iscsi_doorclt.o   \
2025                         iscsi_targetparam.o utils.o kifconf.o

2027 #
2028 #       ntxn 10Gb/1Gb NIC driver module
2029 #
2030 NTXN_OBJS =     unm_nic_init.o unm_gem.o unm_nic_hw.o unm_ndd.o \
2031                         unm_nic_main.o unm_nic_isr.o unm_nic_ctx.o niu.o

2033 #
2034 #       Myricom 10Gb NIC driver module
2035 #
2036 MYRI10GE_OBJS = myri10ge.o myri10ge_lro.o

2038 #       nulldriver module
```



```
2039 #
2040 NULLDRIVER_OBJS =       nulldriver.o

2042 TPM_OBJS =      tpm.o tpm_hcall.o
```

```
**********************************************************
    1065 Thu Jul 11 01:29:49 2013
new/usr/src/uts/common/crypto/io/Makefile
first pass
**********************************************************
```

```
   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 # ident "%Z%%M% %I%     %E% SMI"
  24 #
  25 # Copyright 2003 Sun Microsystems, Inc.  All rights reserved.
  26 # Use is subject to license terms.
  27 #
  28 # uts/common/crypto/io/Makefile
  29 #
  30 # include global definitions
  31 include ../../../../Makefile.master

  33 .KEEP_STATE:

  35 # EXPORT DELETE START
  36 EXPORT_SRC:
  37         $(RM) Makefile+ aes.c+ arcfour.c+ blowfish.c+ dprov.c+ rsa.c+
  38         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  39                 < aes.c > aes.c+
  40         $(MV) aes.c+ aes.c
  41         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  42                 < arcfour.c > arcfour.c+
  43         $(MV) arcfour.c+ arcfour.c
  44         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  45                 < blowfish.c > blowfish.c+
  46         $(MV) blowfish.c+ blowfish.c
  47         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  48                 < dprov.c > dprov.c+
  49         $(MV) dprov.c+ dprov.c
  50         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  51                 < rsa.c > rsa.c+
  52         $(MV) rsa.c+ rsa.c
  53         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
  54                 < Makefile > Makefile+
  55         $(RM) Makefile
  56         $(MV) Makefile+ Makefile
  57         $(CHMOD) 444 Makefile aes.c arcfour.c blowfish.c dprov.c rsa.c

  59 # EXPORT DELETE END
```

```
**********************************************************
   40891 Thu Jul 11 01:29:50 2013
new/usr/src/uts/common/crypto/io/aes.c
first pass
**********************************************************
______unchanged_portion_omitted_

 309 /* EXPORT DELETE START */

 309 /*
 310  * Initialize key schedules for AES
 311  */
 312 static int
 313 init_keysched(crypto_key_t *key, void *newbie)
 314 {
 315         /*
 316          * Only keys by value are supported by this module.
 317          */
 318         switch (key->ck_format) {
 319         case CRYPTO_KEY_RAW:
 320                 if (key->ck_length < AES_MINBITS ||
 321                     key->ck_length > AES_MAXBITS) {
 322                         return (CRYPTO_KEY_SIZE_RANGE);
 323                 }

 325                 /* key length must be either 128, 192, or 256 */
 326                 if ((key->ck_length & 63) != 0)
 327                         return (CRYPTO_KEY_SIZE_RANGE);
 328                 break;
 329         default:
 330                 return (CRYPTO_KEY_TYPE_INCONSISTENT);
 331         }

 333         aes_init_keysched(key->ck_data, key->ck_length, newbie);
 334         return (CRYPTO_SUCCESS);
 335 }

 339 /* EXPORT DELETE END */

 337 /*
 338  * KCF software provider control entry points.
 339  */
 340 /* ARGSUSED */
 341 static void
 342 aes_provider_status(crypto_provider_handle_t provider, uint_t *status)
 343 {
 344         *status = CRYPTO_PROVIDER_READY;
 345 }
______unchanged_portion_omitted_



 363 /*
 364  * KCF software provider encrypt entry points.
 365  */
 366 static int
 367 aes_common_init(crypto_ctx_t *ctx, crypto_mechanism_t *mechanism,
 368     crypto_key_t *key, crypto_spi_ctx_template_t template,
 369     crypto_req_handle_t req, boolean_t is_encrypt_init)
 370 {

 376 /* EXPORT DELETE START */

 371         aes_ctx_t *aes_ctx;
 372         int rv;
 373         int kmflag;
```



```
 375         /*
 376          * Only keys by value are supported by this module.
 377          */
 378         if (key->ck_format != CRYPTO_KEY_RAW) {
 379                 return (CRYPTO_KEY_TYPE_INCONSISTENT);
 380         }

 382         kmflag = crypto_kmflag(req);
 383         if ((rv = aes_check_mech_param(mechanism, &aes_ctx, kmflag))
 384             != CRYPTO_SUCCESS)
 385                 return (rv);

 387         rv = aes_common_init_ctx(aes_ctx, template, mechanism, key, kmflag,
 388             is_encrypt_init);
 389         if (rv != CRYPTO_SUCCESS) {
 390                 crypto_free_mode_ctx(aes_ctx);
 391                 return (rv);
 392         }

 394         ctx->cc_provider_private = aes_ctx;

 403 /* EXPORT DELETE END */

 396         return (CRYPTO_SUCCESS);
 397 }
______unchanged_portion_omitted_


 415 static int
 416 aes_encrypt(crypto_ctx_t *ctx, crypto_data_t *plaintext,
 417     crypto_data_t *ciphertext, crypto_req_handle_t req)
 418 {
 419         int ret = CRYPTO_FAILED;

 430 /* EXPORT DELETE START */

 421         aes_ctx_t *aes_ctx;
 422         size_t saved_length, saved_offset, length_needed;

 424         ASSERT(ctx->cc_provider_private != NULL);
 425         aes_ctx = ctx->cc_provider_private;

 427         /*
 428          * For block ciphers, plaintext must be a multiple of AES block size.
 429          * This test is only valid for ciphers whose blocksize is a power of 2.
 430          */
 431         if (((aes_ctx->ac_flags & (CTR_MODE|CCM_MODE|GCM_MODE|GMAC_MODE))
 432             == 0) && (plaintext->cd_length & (AES_BLOCK_LEN - 1)) != 0)
 433                 return (CRYPTO_DATA_LEN_RANGE);

 435         AES_ARG_INPLACE(plaintext, ciphertext);

 437         /*
 438          * We need to just return the length needed to store the output.
 439          * We should not destroy the context for the following case.
 440          */
 441         switch (aes_ctx->ac_flags & (CCM_MODE|GCM_MODE|GMAC_MODE)) {
 442         case CCM_MODE:
 443                 length_needed = plaintext->cd_length + aes_ctx->ac_mac_len;
 444                 break;
 445         case GCM_MODE:
 446                 length_needed = plaintext->cd_length + aes_ctx->ac_tag_len;
 447                 break;
 448         case GMAC_MODE:
 449                 if (plaintext->cd_length != 0)
```

```
450                         return (CRYPTO_ARGUMENTS_BAD);

452                 length_needed = aes_ctx->ac_tag_len;
453                 break;
454         default:
455                 length_needed = plaintext->cd_length;
456         }

458         if (ciphertext->cd_length < length_needed) {
459                 ciphertext->cd_length = length_needed;
460                 return (CRYPTO_BUFFER_TOO_SMALL);
461         }

463         saved_length = ciphertext->cd_length;
464         saved_offset = ciphertext->cd_offset;

466         /*
467          * Do an update on the specified input data.
468          */
469         ret = aes_encrypt_update(ctx, plaintext, ciphertext, req);
470         if (ret != CRYPTO_SUCCESS) {
471                 return (ret);
472         }

474         /*
475          * For CCM mode, aes_ccm_encrypt_final() will take care of any
476          * left-over unprocessed data, and compute the MAC
477          */
478         if (aes_ctx->ac_flags & CCM_MODE) {
479                 /*
480                  * ccm_encrypt_final() will compute the MAC and append
481                  * it to existing ciphertext. So, need to adjust the left over
482                  * length value accordingly
483                  */

485                 /* order of following 2 lines MUST not be reversed */
486                 ciphertext->cd_offset = ciphertext->cd_length;
487                 ciphertext->cd_length = saved_length - ciphertext->cd_length;
488                 ret = ccm_encrypt_final((ccm_ctx_t *)aes_ctx, ciphertext,
489                     AES_BLOCK_LEN, aes_encrypt_block, aes_xor_block);
490                 if (ret != CRYPTO_SUCCESS) {
491                         return (ret);
492                 }

494                 if (plaintext != ciphertext) {
495                         ciphertext->cd_length =
496                             ciphertext->cd_offset - saved_offset;
497                 }
498                 ciphertext->cd_offset = saved_offset;
499         } else if (aes_ctx->ac_flags & (GCM_MODE|GMAC_MODE)) {
500                 /*
501                  * gcm_encrypt_final() will compute the MAC and append
502                  * it to existing ciphertext. So, need to adjust the left over
503                  * length value accordingly
504                  */

506                 /* order of following 2 lines MUST not be reversed */
507                 ciphertext->cd_offset = ciphertext->cd_length;
508                 ciphertext->cd_length = saved_length - ciphertext->cd_length;
509                 ret = gcm_encrypt_final((gcm_ctx_t *)aes_ctx, ciphertext,
510                     AES_BLOCK_LEN, aes_encrypt_block, aes_copy_block,
511                     aes_xor_block);
512                 if (ret != CRYPTO_SUCCESS) {
513                         return (ret);
514                 }
```



```
516                 if (plaintext != ciphertext) {
517                         ciphertext->cd_length =
518                             ciphertext->cd_offset - saved_offset;
519                 }
520                 ciphertext->cd_offset = saved_offset;
521         }

523         ASSERT(aes_ctx->ac_remainder_len == 0);
524         (void) aes_free_context(ctx);

537 /* EXPORT DELETE END */

526         return (ret);
527 }


530 static int
531 aes_decrypt(crypto_ctx_t *ctx, crypto_data_t *ciphertext,
532     crypto_data_t *plaintext, crypto_req_handle_t req)
533 {
534         int ret = CRYPTO_FAILED;

549 /* EXPORT DELETE START */

536         aes_ctx_t *aes_ctx;
537         off_t saved_offset;
538         size_t saved_length, length_needed;

540         ASSERT(ctx->cc_provider_private != NULL);
541         aes_ctx = ctx->cc_provider_private;

543         /*
544          * For block ciphers, plaintext must be a multiple of AES block size.
545          * This test is only valid for ciphers whose blocksize is a power of 2.
546          */
547         if (((aes_ctx->ac_flags & (CTR_MODE|CCM_MODE|GCM_MODE|GMAC_MODE))
548             == 0) && (ciphertext->cd_length & (AES_BLOCK_LEN - 1)) != 0) {
549                 return (CRYPTO_ENCRYPTED_DATA_LEN_RANGE);
550         }

552         AES_ARG_INPLACE(ciphertext, plaintext);

554         /*
555          * Return length needed to store the output.
556          * Do not destroy context when plaintext buffer is too small.
557          *
558          * CCM:  plaintext is MAC len smaller than cipher text
559          * GCM:  plaintext is TAG len smaller than cipher text
560          * GMAC: plaintext length must be zero
561          */
562         switch (aes_ctx->ac_flags & (CCM_MODE|GCM_MODE|GMAC_MODE)) {
563         case CCM_MODE:
564                 length_needed = aes_ctx->ac_processed_data_len;
565                 break;
566         case GCM_MODE:
567                 length_needed = ciphertext->cd_length - aes_ctx->ac_tag_len;
568                 break;
569         case GMAC_MODE:
570                 if (plaintext->cd_length != 0)
571                         return (CRYPTO_ARGUMENTS_BAD);

573                 length_needed = 0;
574                 break;
575         default:
576                 length_needed = ciphertext->cd_length;
577         }
```

```
579         if (plaintext->cd_length < length_needed) {
580                 plaintext->cd_length = length_needed;
581                 return (CRYPTO_BUFFER_TOO_SMALL);
582         }

584         saved_offset = plaintext->cd_offset;
585         saved_length = plaintext->cd_length;

587         /*
588          * Do an update on the specified input data.
589          */
590         ret = aes_decrypt_update(ctx, ciphertext, plaintext, req);
591         if (ret != CRYPTO_SUCCESS) {
592                 goto cleanup;
593         }

595         if (aes_ctx->ac_flags & CCM_MODE) {
596                 ASSERT(aes_ctx->ac_processed_data_len == aes_ctx->ac_data_len);
597                 ASSERT(aes_ctx->ac_processed_mac_len == aes_ctx->ac_mac_len);

599                 /* order of following 2 lines MUST not be reversed */
600                 plaintext->cd_offset = plaintext->cd_length;
601                 plaintext->cd_length = saved_length - plaintext->cd_length;

603                 ret = ccm_decrypt_final((ccm_ctx_t *)aes_ctx, plaintext,
604                     AES_BLOCK_LEN, aes_encrypt_block, aes_copy_block,
605                     aes_xor_block);
606                 if (ret == CRYPTO_SUCCESS) {
607                         if (plaintext != ciphertext) {
608                                 plaintext->cd_length =
609                                     plaintext->cd_offset - saved_offset;
610                         }
611                 } else {
612                         plaintext->cd_length = saved_length;
613                 }

615                 plaintext->cd_offset = saved_offset;
616         } else if (aes_ctx->ac_flags & (GCM_MODE|GMAC_MODE)) {
617                 /* order of following 2 lines MUST not be reversed */
618                 plaintext->cd_offset = plaintext->cd_length;
619                 plaintext->cd_length = saved_length - plaintext->cd_length;

621                 ret = gcm_decrypt_final((gcm_ctx_t *)aes_ctx, plaintext,
622                     AES_BLOCK_LEN, aes_encrypt_block, aes_xor_block);
623                 if (ret == CRYPTO_SUCCESS) {
624                         if (plaintext != ciphertext) {
625                                 plaintext->cd_length =
626                                     plaintext->cd_offset - saved_offset;
627                         }
628                 } else {
629                         plaintext->cd_length = saved_length;
630                 }

632                 plaintext->cd_offset = saved_offset;
633         }

635         ASSERT(aes_ctx->ac_remainder_len == 0);

637 cleanup:
638         (void) aes_free_context(ctx);

655 /* EXPORT DELETE END */

640         return (ret);
641 }
```

*unchanged_portion_omitted_*


```
806 /* ARGSUSED */
807 static int
808 aes_encrypt_final(crypto_ctx_t *ctx, crypto_data_t *data,
809     crypto_req_handle_t req)
810 {

829 /* EXPORT DELETE START */

811         aes_ctx_t *aes_ctx;
812         int ret;

814         ASSERT(ctx->cc_provider_private != NULL);
815         aes_ctx = ctx->cc_provider_private;

817         if (data->cd_format != CRYPTO_DATA_RAW &&
818             data->cd_format != CRYPTO_DATA_UIO &&
819             data->cd_format != CRYPTO_DATA_MBLK) {
820                 return (CRYPTO_ARGUMENTS_BAD);
821         }

823         if (aes_ctx->ac_flags & CTR_MODE) {
824                 if (aes_ctx->ac_remainder_len > 0) {
825                         ret = ctr_mode_final((ctr_ctx_t *)aes_ctx, data,
826                             aes_encrypt_block);
827                         if (ret != CRYPTO_SUCCESS)
828                                 return (ret);
829                 }
830         } else if (aes_ctx->ac_flags & CCM_MODE) {
831                 ret = ccm_encrypt_final((ccm_ctx_t *)aes_ctx, data,
832                     AES_BLOCK_LEN, aes_encrypt_block, aes_xor_block);
833                 if (ret != CRYPTO_SUCCESS) {
834                         return (ret);
835                 }
836         } else if (aes_ctx->ac_flags & (GCM_MODE|GMAC_MODE)) {
837                 size_t saved_offset = data->cd_offset;

839                 ret = gcm_encrypt_final((gcm_ctx_t *)aes_ctx, data,
840                     AES_BLOCK_LEN, aes_encrypt_block, aes_copy_block,
841                     aes_xor_block);
842                 if (ret != CRYPTO_SUCCESS) {
843                         return (ret);
844                 }
845                 data->cd_length = data->cd_offset - saved_offset;
846                 data->cd_offset = saved_offset;
847         } else {
848                 /*
849                  * There must be no unprocessed plaintext.
850                  * This happens if the length of the last data is
851                  * not a multiple of the AES block length.
852                  */
853                 if (aes_ctx->ac_remainder_len > 0) {
854                         return (CRYPTO_DATA_LEN_RANGE);
855                 }
856                 data->cd_length = 0;
857         }

859         (void) aes_free_context(ctx);

881 /* EXPORT DELETE END */

861         return (CRYPTO_SUCCESS);
862 }

864 /* ARGSUSED */
865 static int
```

```
 866 aes_decrypt_final(crypto_ctx_t *ctx, crypto_data_t *data,
 867     crypto_req_handle_t req)
 868 {
 892 /* EXPORT DELETE START */
 869         aes_ctx_t *aes_ctx;
 870         int ret;
 871         off_t saved_offset;
 872         size_t saved_length;

 874         ASSERT(ctx->cc_provider_private != NULL);
 875         aes_ctx = ctx->cc_provider_private;

 877         if (data->cd_format != CRYPTO_DATA_RAW &&
 878             data->cd_format != CRYPTO_DATA_UIO &&
 879             data->cd_format != CRYPTO_DATA_MBLK) {
 880                 return (CRYPTO_ARGUMENTS_BAD);
 881         }

 883         /*
 884          * There must be no unprocessed ciphertext.
 885          * This happens if the length of the last ciphertext is
 886          * not a multiple of the AES block length.
 887          */
 888         if (aes_ctx->ac_remainder_len > 0) {
 889                 if ((aes_ctx->ac_flags & CTR_MODE) == 0)
 890                         return (CRYPTO_ENCRYPTED_DATA_LEN_RANGE);
 891                 else {
 892                         ret = ctr_mode_final((ctr_ctx_t *)aes_ctx, data,
 893                             aes_encrypt_block);
 894                         if (ret == CRYPTO_DATA_LEN_RANGE)
 895                                 ret = CRYPTO_ENCRYPTED_DATA_LEN_RANGE;
 896                         if (ret != CRYPTO_SUCCESS)
 897                                 return (ret);
 898                 }
 899         }

 901         if (aes_ctx->ac_flags & CCM_MODE) {
 902                 /*
 903                  * This is where all the plaintext is returned, make sure
 904                  * the plaintext buffer is big enough
 905                  */
 906                 size_t pt_len = aes_ctx->ac_data_len;
 907                 if (data->cd_length < pt_len) {
 908                         data->cd_length = pt_len;
 909                         return (CRYPTO_BUFFER_TOO_SMALL);
 910                 }

 912                 ASSERT(aes_ctx->ac_processed_data_len == pt_len);
 913                 ASSERT(aes_ctx->ac_processed_mac_len == aes_ctx->ac_mac_len);
 914                 saved_offset = data->cd_offset;
 915                 saved_length = data->cd_length;
 916                 ret = ccm_decrypt_final((ccm_ctx_t *)aes_ctx, data,
 917                     AES_BLOCK_LEN, aes_encrypt_block, aes_copy_block,
 918                     aes_xor_block);
 919                 if (ret == CRYPTO_SUCCESS) {
 920                         data->cd_length = data->cd_offset - saved_offset;
 921                 } else {
 922                         data->cd_length = saved_length;
 923                 }

 925                 data->cd_offset = saved_offset;
 926                 if (ret != CRYPTO_SUCCESS) {
 927                         return (ret);
 928                 }
```



```
 929         } else if (aes_ctx->ac_flags & (GCM_MODE|GMAC_MODE)) {
 930                 /*
 931                  * This is where all the plaintext is returned, make sure
 932                  * the plaintext buffer is big enough
 933                  */
 934                 gcm_ctx_t *ctx = (gcm_ctx_t *)aes_ctx;
 935                 size_t pt_len = ctx->gcm_processed_data_len - ctx->gcm_tag_len;

 937                 if (data->cd_length < pt_len) {
 938                         data->cd_length = pt_len;
 939                         return (CRYPTO_BUFFER_TOO_SMALL);
 940                 }

 942                 saved_offset = data->cd_offset;
 943                 saved_length = data->cd_length;
 944                 ret = gcm_decrypt_final((gcm_ctx_t *)aes_ctx, data,
 945                     AES_BLOCK_LEN, aes_encrypt_block, aes_xor_block);
 946                 if (ret == CRYPTO_SUCCESS) {
 947                         data->cd_length = data->cd_offset - saved_offset;
 948                 } else {
 949                         data->cd_length = saved_length;
 950                 }

 952                 data->cd_offset = saved_offset;
 953                 if (ret != CRYPTO_SUCCESS) {
 954                         return (ret);
 955                 }
 956         }


 959         if ((aes_ctx->ac_flags & (CTR_MODE|CCM_MODE|GCM_MODE|GMAC_MODE)) == 0) {
 960                 data->cd_length = 0;
 961         }

 963         (void) aes_free_context(ctx);
 990 /* EXPORT DELETE END */
 965         return (CRYPTO_SUCCESS);
 966 }
_____unchanged_portion_omitted_

1261 /*
1262  * KCF software provider context template entry points.
1263  */
1264 /* ARGSUSED */
1265 static int
1266 aes_create_ctx_template(crypto_provider_handle_t provider,
1267     crypto_mechanism_t *mechanism, crypto_key_t *key,
1268     crypto_spi_ctx_template_t *tmpl, size_t *tmpl_size, crypto_req_handle_t req)
1269 {
1298 /* EXPORT DELETE START */
1270         void *keysched;
1271         size_t size;
1272         int rv;

1274         if (mechanism->cm_type != AES_ECB_MECH_INFO_TYPE &&
1275             mechanism->cm_type != AES_CBC_MECH_INFO_TYPE &&
1276             mechanism->cm_type != AES_CTR_MECH_INFO_TYPE &&
1277             mechanism->cm_type != AES_CCM_MECH_INFO_TYPE &&
1278             mechanism->cm_type != AES_GCM_MECH_INFO_TYPE &&
1279             mechanism->cm_type != AES_GMAC_MECH_INFO_TYPE)
1280                 return (CRYPTO_MECHANISM_INVALID);
```

```
1282         if ((keysched = aes_alloc_keysched(&size,
1283             crypto_kmflag(req))) == NULL) {
1284                 return (CRYPTO_HOST_MEMORY);
1285         }

1287         /*
1288          * Initialize key schedule.  Key length information is stored
1289          * in the key.
1290          */
1291         if ((rv = init_keysched(key, keysched)) != CRYPTO_SUCCESS) {
1292                 bzero(keysched, size);
1293                 kmem_free(keysched, size);
1294                 return (rv);
1295         }

1297         *tmpl = keysched;
1298         *tmpl_size = size;

1330 /* EXPORT DELETE END */

1300         return (CRYPTO_SUCCESS);
1301 }


1304 static int
1305 aes_free_context(crypto_ctx_t *ctx)
1306 {

1340 /* EXPORT DELETE START */

1307         aes_ctx_t *aes_ctx = ctx->cc_provider_private;

1309         if (aes_ctx != NULL) {
1310                 if (aes_ctx->ac_flags & PROVIDER_OWNS_KEY_SCHEDULE) {
1311                         ASSERT(aes_ctx->ac_keysched_len != 0);
1312                         bzero(aes_ctx->ac_keysched, aes_ctx->ac_keysched_len);
1313                         kmem_free(aes_ctx->ac_keysched,
1314                             aes_ctx->ac_keysched_len);
1315                 }
1316                 crypto_free_mode_ctx(aes_ctx);
1317                 ctx->cc_provider_private = NULL;
1318         }

1355 /* EXPORT DELETE END */

1320         return (CRYPTO_SUCCESS);
1321 }


1324 static int
1325 aes_common_init_ctx(aes_ctx_t *aes_ctx, crypto_spi_ctx_template_t *template,
1326     crypto_mechanism_t *mechanism, crypto_key_t *key, int kmflag,
1327     boolean_t is_encrypt_init)
1328 {
1329         int rv = CRYPTO_SUCCESS;

1368 /* EXPORT DELETE START */

1330         void *keysched;
1331         size_t size;

1333         if (template == NULL) {
1334                 if ((keysched = aes_alloc_keysched(&size, kmflag)) == NULL)
1335                         return (CRYPTO_HOST_MEMORY);
1336                 /*
1337                  * Initialize key schedule.
```



```
1338                  * Key length is stored in the key.
1339                  */
1340                 if ((rv = init_keysched(key, keysched)) != CRYPTO_SUCCESS) {
1341                         kmem_free(keysched, size);
1342                         return (rv);
1343                 }

1345                 aes_ctx->ac_flags |= PROVIDER_OWNS_KEY_SCHEDULE;
1346                 aes_ctx->ac_keysched_len = size;
1347         } else {
1348                 keysched = template;
1349         }
1350         aes_ctx->ac_keysched = keysched;

1352         switch (mechanism->cm_type) {
1353         case AES_CBC_MECH_INFO_TYPE:
1354                 rv = cbc_init_ctx((cbc_ctx_t *)aes_ctx, mechanism->cm_param,
1355                     mechanism->cm_param_len, AES_BLOCK_LEN, aes_copy_block64);
1356                 break;
1357         case AES_CTR_MECH_INFO_TYPE: {
1358                 CK_AES_CTR_PARAMS *pp;

1360                 if (mechanism->cm_param == NULL ||
1361                     mechanism->cm_param_len != sizeof (CK_AES_CTR_PARAMS)) {
1362                         return (CRYPTO_MECHANISM_PARAM_INVALID);
1363                 }
1364                 pp = (CK_AES_CTR_PARAMS *)(void *)mechanism->cm_param;
1365                 rv = ctr_init_ctx((ctr_ctx_t *)aes_ctx, pp->ulCounterBits,
1366                     pp->cb, aes_copy_block);
1367                 break;
1368         }
1369         case AES_CCM_MECH_INFO_TYPE:
1370                 if (mechanism->cm_param == NULL ||
1371                     mechanism->cm_param_len != sizeof (CK_AES_CCM_PARAMS)) {
1372                         return (CRYPTO_MECHANISM_PARAM_INVALID);
1373                 }
1374                 rv = ccm_init_ctx((ccm_ctx_t *)aes_ctx, mechanism->cm_param,
1375                     kmflag, is_encrypt_init, AES_BLOCK_LEN, aes_encrypt_block,
1376                     aes_xor_block);
1377                 break;
1378         case AES_GCM_MECH_INFO_TYPE:
1379                 if (mechanism->cm_param == NULL ||
1380                     mechanism->cm_param_len != sizeof (CK_AES_GCM_PARAMS)) {
1381                         return (CRYPTO_MECHANISM_PARAM_INVALID);
1382                 }
1383                 rv = gcm_init_ctx((gcm_ctx_t *)aes_ctx, mechanism->cm_param,
1384                     AES_BLOCK_LEN, aes_encrypt_block, aes_copy_block,
1385                     aes_xor_block);
1386                 break;
1387         case AES_GMAC_MECH_INFO_TYPE:
1388                 if (mechanism->cm_param == NULL ||
1389                     mechanism->cm_param_len != sizeof (CK_AES_GMAC_PARAMS)) {
1390                         return (CRYPTO_MECHANISM_PARAM_INVALID);
1391                 }
1392                 rv = gmac_init_ctx((gcm_ctx_t *)aes_ctx, mechanism->cm_param,
1393                     AES_BLOCK_LEN, aes_encrypt_block, aes_copy_block,
1394                     aes_xor_block);
1395                 break;
1396         case AES_ECB_MECH_INFO_TYPE:
1397                 aes_ctx->ac_flags |= ECB_MODE;
1398         }

1400         if (rv != CRYPTO_SUCCESS) {
1401                 if (aes_ctx->ac_flags & PROVIDER_OWNS_KEY_SCHEDULE) {
1402                         bzero(keysched, size);
1403                         kmem_free(keysched, size);
```

```
1404                 }
1405         }
```

```
1447 /* EXPORT DELETE END */
```

```
1407         return (rv);
1408 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   14910 Thu Jul 11 01:29:50 2013
new/usr/src/uts/common/crypto/io/arcfour.c
first pass
**********************************************************
______unchanged_portion_omitted_

 198 /* ARGSUSED */
 199 static int
 200 rc4_common_init(crypto_ctx_t *ctx, crypto_mechanism_t *mechanism,
 201     crypto_key_t *key, crypto_spi_ctx_template_t template,
 202     crypto_req_handle_t req)
 203 {
 205 /* EXPORT DELETE START */
 204         ARCFour_key *keystream;

 206         if ((mechanism)->cm_type != RC4_MECH_INFO_TYPE)
 207                 return (CRYPTO_MECHANISM_INVALID);

 209         if (key->ck_format != CRYPTO_KEY_RAW)
 210                 return (CRYPTO_KEY_TYPE_INCONSISTENT);

 212         if (key->ck_length < ARCFOUR_MIN_KEY_BITS ||
 213             key->ck_length > ARCFOUR_MAX_KEY_BITS) {
 214                 return (CRYPTO_KEY_SIZE_RANGE);
 215         }

 217         /*
 218          * Allocate an RC4 key stream.
 219          */
 220         if ((keystream = kmem_alloc(sizeof (ARCFour_key),
 221             crypto_kmflag(req))) == NULL)
 222                 return (CRYPTO_HOST_MEMORY);

 224         arcfour_key_init(keystream, key->ck_data,
 225             CRYPTO_BITS2BYTES(key->ck_length));

 227         ctx->cc_provider_private = keystream;
 232 /* EXPORT DELETE END */
 229         return (CRYPTO_SUCCESS);
 230 }
______unchanged_portion_omitted_

 246 /* ARGSUSED */
 247 static int
 248 rc4_crypt_update(crypto_ctx_t *ctx, crypto_data_t *input, crypto_data_t *output,
 249     crypto_req_handle_t req)
 250 {
 251         int ret = CRYPTO_SUCCESS;
 258 /* EXPORT DELETE START */
 253         ARCFour_key *key;
 254         off_t saveoffset;

 256         ASSERT(ctx->cc_provider_private != NULL);

 258         if ((ctx->cc_flags & CRYPTO_USE_OPSTATE) && ctx->cc_opstate != NULL)
 259                 key = ctx->cc_opstate;
 260         else
 261                 key = ctx->cc_provider_private;
```



```
 263         /* Simple case: in-line encipherment */

 265         if (output == NULL) {
 266                 switch (input->cd_format) {
 267                 case CRYPTO_DATA_RAW: {
 268                         char *start, *end;
 269                         start = input->cd_raw.iov_base + input->cd_offset;

 271                         end =  input->cd_raw.iov_base + input->cd_raw.iov_len;

 273                         if (start + input->cd_length > end)
 274                                 return (CRYPTO_DATA_INVALID);

 276                         arcfour_crypt(key, (uchar_t *)start, (uchar_t *)start,
 277                             input->cd_length);
 278                         break;
 279                 }
 280                 case CRYPTO_DATA_MBLK: {
 281                         uchar_t *start, *end;
 282                         size_t len, left;
 283                         mblk_t *mp = input->cd_mp, *mp1, *mp2;

 285                         ASSERT(mp != NULL);

 287                         mp1 = advance_position(mp, input->cd_offset, &start);

 289                         if (mp1 == NULL)
 290                                 return (CRYPTO_DATA_LEN_RANGE);

 292                         mp2 = advance_position(mp, input->cd_offset +
 293                             input->cd_length, &end);

 295                         if (mp2 == NULL)
 296                                 return (CRYPTO_DATA_LEN_RANGE);

 298                         left = input->cd_length;
 299                         while (mp1 != NULL) {
 300                                 if (_PTRDIFF(mp1->b_wptr, start) > left) {
 301                                         len = left;
 302                                         arcfour_crypt(key, start, start, len);
 303                                         mp1 = NULL;
 304                                 } else {
 305                                         len = _PTRDIFF(mp1->b_wptr, start);
 306                                         arcfour_crypt(key, start, start, len);
 307                                         mp1 = mp1->b_cont;
 308                                         start = mp1->b_rptr;
 309                                         left -= len;
 310                                 }
 311                         }
 312                         break;
 313                 }
 314                 case CRYPTO_DATA_UIO: {
 315                         uio_t *uiop = input->cd_uio;
 316                         off_t offset = input->cd_offset;
 317                         size_t length = input->cd_length;
 318                         uint_t vec_idx;
 319                         size_t cur_len;

 321                         /*
 322                          * Jump to the first iovec containing data to be
 323                          * processed.
 324                          */
 325                         for (vec_idx = 0; vec_idx < uiop->uio_iovcnt &&
 326                             offset >= uiop->uio_iov[vec_idx].iov_len;
 327                             offset -= uiop->uio_iov[vec_idx++].iov_len)
 328                                 ;
```

```
329                         if (vec_idx == uiop->uio_iovcnt) {
330                                 return (CRYPTO_DATA_LEN_RANGE);
331                         }

333                         /*
334                          * Now process the iovecs.
335                          */
336                         while (vec_idx < uiop->uio_iovcnt && length > 0) {
337                                 uchar_t *start;
338                                 iovec_t *iovp = &(uiop->uio_iov[vec_idx]);

340                                 cur_len = MIN(iovp->iov_len - offset, length);

342                                 start = (uchar_t *)(iovp->iov_base + offset);
343                                 arcfour_crypt(key, start + offset,
344                                     start + offset, cur_len);

346                                 length -= cur_len;
347                                 vec_idx++;
348                                 offset = 0;
349                         }

351                         if (vec_idx == uiop->uio_iovcnt && length > 0) {

353                                 return (CRYPTO_DATA_LEN_RANGE);
354                         }
355                         break;
356                 }
357                 }
358                 return (CRYPTO_SUCCESS);
359         }

361         /*
362          * We need to just return the length needed to store the output.
363          * We should not destroy the context for the following case.
364          */

366         if (input->cd_length > output->cd_length) {
367                 output->cd_length = input->cd_length;
368                 return (CRYPTO_BUFFER_TOO_SMALL);
369         }

371         saveoffset = output->cd_offset;

373         switch (input->cd_format) {
374         case CRYPTO_DATA_RAW: {
375                 char *start, *end;
376                 start = input->cd_raw.iov_base + input->cd_offset;

378                 end =  input->cd_raw.iov_base + input->cd_raw.iov_len;

380                 if (start + input->cd_length > end)
381                         return (CRYPTO_DATA_LEN_RANGE);

383                 ret = crypto_arcfour_crypt(key, (uchar_t *)start, output,
384                     input->cd_length);

386                 if (ret != CRYPTO_SUCCESS)
387                         return (ret);
388                 break;
389         }
390         case CRYPTO_DATA_MBLK: {
391                 uchar_t *start, *end;
392                 size_t len, left;
393                 mblk_t *mp = input->cd_mp, *mp1, *mp2;
```



```
395                 ASSERT(mp != NULL);

397                 mp1 = advance_position(mp, input->cd_offset, &start);

399                 if (mp1 == NULL)
400                         return (CRYPTO_DATA_LEN_RANGE);

402                 mp2 = advance_position(mp, input->cd_offset + input->cd_length,
403                     &end);

405                 if (mp2 == NULL)
406                         return (CRYPTO_DATA_LEN_RANGE);

408                 left = input->cd_length;
409                 while (mp1 != NULL) {
410                         if (_PTRDIFF(mp1->b_wptr, start) > left) {
411                                 len = left;
412                                 ret = crypto_arcfour_crypt(key, start, output,
413                                     len);
414                                 if (ret != CRYPTO_SUCCESS)
415                                         return (ret);
416                                 mp1 = NULL;
417                         } else {
418                                 len = _PTRDIFF(mp1->b_wptr, start);
419                                 ret = crypto_arcfour_crypt(key, start, output,
420                                     len);
421                                 if (ret != CRYPTO_SUCCESS)
422                                         return (ret);
423                                 mp1 = mp1->b_cont;
424                                 start = mp1->b_rptr;
425                                 left -= len;
426                                 output->cd_offset += len;
427                         }
428                 }
429                 break;
430         }
431         case CRYPTO_DATA_UIO: {
432                 uio_t *uiop = input->cd_uio;
433                 off_t offset = input->cd_offset;
434                 size_t length = input->cd_length;
435                 uint_t vec_idx;
436                 size_t cur_len;

438                 /*
439                  * Jump to the first iovec containing data to be
440                  * processed.
441                  */
442                 for (vec_idx = 0; vec_idx < uiop->uio_iovcnt &&
443                     offset >= uiop->uio_iov[vec_idx].iov_len;
444                     offset -= uiop->uio_iov[vec_idx++].iov_len)
445                         ;
446                 if (vec_idx == uiop->uio_iovcnt) {
447                         return (CRYPTO_DATA_LEN_RANGE);
448                 }

450                 /*
451                  * Now process the iovecs.
452                  */
453                 while (vec_idx < uiop->uio_iovcnt && length > 0) {
454                         uchar_t *start;
455                         iovec_t *iovp = &(uiop->uio_iov[vec_idx]);
456                         cur_len = MIN(iovp->iov_len - offset, length);

458                         start = (uchar_t *)(iovp->iov_base + offset);
459                         ret = crypto_arcfour_crypt(key, start + offset,
460                             output, cur_len);
```

```
                        if (ret != CRYPTO_SUCCESS)
                                return (ret);

                        length -= cur_len;
                        vec_idx++;
                        offset = 0;
                        output->cd_offset += cur_len;
                }

                if (vec_idx == uiop->uio_iovcnt && length > 0) {

                        return (CRYPTO_DATA_LEN_RANGE);
                }
        }
        }

        output->cd_offset = saveoffset;
        output->cd_length = input->cd_length;

/* EXPORT DELETE END */

        return (ret);
}
```

_____unchanged_portion_omitted_

```
/* ARGSUSED */
static int
rc4_free_context(crypto_ctx_t *ctx)
{

/* EXPORT DELETE START */

        ARCFour_key *keystream = ctx->cc_provider_private;

        if (keystream != NULL) {
                bzero(keystream, sizeof (ARCFour_key));
                kmem_free(keystream, sizeof (ARCFour_key));
                ctx->cc_provider_private = NULL;
        }

/* EXPORT DELETE END */

        return (CRYPTO_SUCCESS);
}
```

_____unchanged_portion_omitted_

```
**********************************************************
   22809 Thu Jul 11 01:29:51 2013
new/usr/src/uts/common/crypto/io/blowfish.c
first pass
**********************************************************
______unchanged_portion_omitted_

 237 /*
 238  * Initialize key schedules for blowfish
 239  */
 240 static int
 241 init_keysched(crypto_key_t *key, void *keysched)
 242 {
 243 /* EXPORT DELETE START */
 243         /*
 244          * Only keys by value are supported by this module.
 245          */
 246         switch (key->ck_format) {
 247         case CRYPTO_KEY_RAW:
 248                 if (key->ck_length < BLOWFISH_MINBITS ||
 249                     key->ck_length > BLOWFISH_MAXBITS) {
 250                         return (CRYPTO_KEY_SIZE_RANGE);
 251                 }
 252                 break;
 253         default:
 254                 return (CRYPTO_KEY_TYPE_INCONSISTENT);
 255         }

 257         blowfish_init_keysched(key->ck_data, key->ck_length, keysched);
 259 /* EXPORT DELETE END */
 258         return (CRYPTO_SUCCESS);
 259 }
______unchanged_portion_omitted_

 271 /*
 272  * KCF software provider encrypt entry points.
 273  */
 274 static int
 275 blowfish_common_init(crypto_ctx_t *ctx, crypto_mechanism_t *mechanism,
 276     crypto_key_t *key, crypto_spi_ctx_template_t template,
 277     crypto_req_handle_t req)
 278 {
 282 /* EXPORT DELETE START */
 279         blowfish_ctx_t *blowfish_ctx;
 280         int rv;
 281         int kmflag;

 283         /*
 284          * Only keys by value are supported by this module.
 285          */
 286         if (key->ck_format != CRYPTO_KEY_RAW) {
 287                 return (CRYPTO_KEY_TYPE_INCONSISTENT);
 288         }

 290         if (!BLOWFISH_VALID_MECH(mechanism))
 291                 return (CRYPTO_MECHANISM_INVALID);

 293         if (mechanism->cm_param != NULL &&
 294             mechanism->cm_param_len != BLOWFISH_BLOCK_LEN)
 295                 return (CRYPTO_MECHANISM_PARAM_INVALID);

 297         kmflag = crypto_kmflag(req);
 298         switch (mechanism->cm_type) {
 299         case BLOWFISH_ECB_MECH_INFO_TYPE:
```



```
 300                 blowfish_ctx = ecb_alloc_ctx(kmflag);
 301                 break;
 302         case BLOWFISH_CBC_MECH_INFO_TYPE:
 303                 blowfish_ctx = cbc_alloc_ctx(kmflag);
 304                 break;
 305         }
 306         if (blowfish_ctx == NULL)
 307                 return (CRYPTO_HOST_MEMORY);

 309         rv = blowfish_common_init_ctx(blowfish_ctx, template, mechanism,
 310             key, kmflag);
 311         if (rv != CRYPTO_SUCCESS) {
 312                 crypto_free_mode_ctx(blowfish_ctx);
 313                 return (rv);
 314         }

 316         ctx->cc_provider_private = blowfish_ctx;

 323 /* EXPORT DELETE END */

 318         return (CRYPTO_SUCCESS);
 319 }
______unchanged_portion_omitted_

 334 /* ARGSUSED */
 335 static int
 336 blowfish_encrypt(crypto_ctx_t *ctx, crypto_data_t *plaintext,
 337     crypto_data_t *ciphertext, crypto_req_handle_t req)
 338 {
 339         int ret;

 348 /* EXPORT DELETE START */
 341         blowfish_ctx_t *blowfish_ctx;

 343         /*
 344          * Plaintext must be a multiple of blowfish block size.
 345          * This test only works for non-padded mechanisms
 346          * when blocksize is 2^N.
 347          */
 348         if ((plaintext->cd_length & (BLOWFISH_BLOCK_LEN - 1)) != 0)
 349                 return (CRYPTO_DATA_LEN_RANGE);

 351         ASSERT(ctx->cc_provider_private != NULL);
 352         blowfish_ctx = ctx->cc_provider_private;

 354         BLOWFISH_ARG_INPLACE(plaintext, ciphertext);

 356         /*
 357          * We need to just return the length needed to store the output.
 358          * We should not destroy the context for the following case.
 359          */
 360         if (ciphertext->cd_length < plaintext->cd_length) {
 361                 ciphertext->cd_length = plaintext->cd_length;
 362                 return (CRYPTO_BUFFER_TOO_SMALL);
 363         }

 365         /*
 366          * Do an update on the specified input data.
 367          */
 368         ret = blowfish_encrypt_update(ctx, plaintext, ciphertext, req);
 369         ASSERT(blowfish_ctx->bc_remainder_len  == 0);
 370         (void) blowfish_free_context(ctx);

 381 /* EXPORT DELETE END */
```

```
372         /* LINTED */
373         return (ret);
374 }

376 /* ARGSUSED */
377 static int
378 blowfish_decrypt(crypto_ctx_t *ctx, crypto_data_t *ciphertext,
379     crypto_data_t *plaintext, crypto_req_handle_t req)
380 {
381         int ret;

394 /* EXPORT DELETE START */

383         blowfish_ctx_t *blowfish_ctx;

385         /*
386          * Ciphertext must be a multiple of blowfish block size.
387          * This test only works for non-padded mechanisms
388          * when blocksize is 2^N.
389          */
390         if ((ciphertext->cd_length & (BLOWFISH_BLOCK_LEN - 1)) != 0)
391                 return (CRYPTO_ENCRYPTED_DATA_LEN_RANGE);

393         ASSERT(ctx->cc_provider_private != NULL);
394         blowfish_ctx = ctx->cc_provider_private;

396         BLOWFISH_ARG_INPLACE(ciphertext, plaintext);

398         /*
399          * We need to just return the length needed to store the output.
400          * We should not destroy the context for the following case.
401          */
402         if (plaintext->cd_length < ciphertext->cd_length) {
403                 plaintext->cd_length = ciphertext->cd_length;
404                 return (CRYPTO_BUFFER_TOO_SMALL);
405         }

407         /*
408          * Do an update on the specified input data.
409          */
410         ret = blowfish_decrypt_update(ctx, ciphertext, plaintext, req);
411         ASSERT(blowfish_ctx->bc_remainder_len == 0);
412         (void) blowfish_free_context(ctx);

427 /* EXPORT DELETE END */

414         /* LINTED */
415         return (ret);
416 }
```

unchanged_portion_omitted_

```
544 /* ARGSUSED */
545 static int
546 blowfish_encrypt_final(crypto_ctx_t *ctx, crypto_data_t *data,
547     crypto_req_handle_t req)
548 {

565 /* EXPORT DELETE START */

549         blowfish_ctx_t *blowfish_ctx;

551         ASSERT(ctx->cc_provider_private != NULL);
552         blowfish_ctx = ctx->cc_provider_private;

554         /*
555          * There must be no unprocessed data.
```



```
556          * This happens if the length of the last data is
557          * not a multiple of the BLOWFISH block length.
558          */
559         if (blowfish_ctx->bc_remainder_len > 0)
560                 return (CRYPTO_DATA_LEN_RANGE);

562         (void) blowfish_free_context(ctx);
563         data->cd_length = 0;

583 /* EXPORT DELETE END */

565         return (CRYPTO_SUCCESS);
566 }

568 /* ARGSUSED */
569 static int
570 blowfish_decrypt_final(crypto_ctx_t *ctx, crypto_data_t *data,
571     crypto_req_handle_t req)
572 {

594 /* EXPORT DELETE START */

573         blowfish_ctx_t *blowfish_ctx;

575         ASSERT(ctx->cc_provider_private != NULL);
576         blowfish_ctx = ctx->cc_provider_private;

578         /*
579          * There must be no unprocessed ciphertext.
580          * This happens if the length of the last ciphertext is
581          * not a multiple of the BLOWFISH block length.
582          */
583         if (blowfish_ctx->bc_remainder_len > 0)
584                 return (CRYPTO_ENCRYPTED_DATA_LEN_RANGE);

586         (void) blowfish_free_context(ctx);
587         data->cd_length = 0;

612 /* EXPORT DELETE END */

589         return (CRYPTO_SUCCESS);
590 }
```

unchanged_portion_omitted_

```
766 /*
767  * KCF software provider context template entry points.
768  */
769 /* ARGSUSED */
770 static int
771 blowfish_create_ctx_template(crypto_provider_handle_t provider,
772     crypto_mechanism_t *mechanism, crypto_key_t *key,
773     crypto_spi_ctx_template_t *tmpl, size_t *tmpl_size, crypto_req_handle_t req)
774 {

801 /* EXPORT DELETE START */

775         void *keysched;
776         size_t size;
777         int rv;

779         if (!BLOWFISH_VALID_MECH(mechanism))
780                 return (CRYPTO_MECHANISM_INVALID);

782         if ((keysched = blowfish_alloc_keysched(&size,
783             crypto_kmflag(req))) == NULL) {
784                 return (CRYPTO_HOST_MEMORY);
```

```
785         }

787         /*
788          * Initialize key schedule.  Key length information is stored
789          * in the key.
790          */
791         if ((rv = init_keysched(key, keysched)) != CRYPTO_SUCCESS) {
792                 bzero(keysched, size);
793                 kmem_free(keysched, size);
794                 return (rv);
795         }

797         *tmpl = keysched;
798         *tmpl_size = size;

828 /* EXPORT DELETE END */

800         return (CRYPTO_SUCCESS);
801 }
```

_____unchanged_portion_omitted_

```
824 /* ARGSUSED */
825 static int
826 blowfish_common_init_ctx(blowfish_ctx_t *blowfish_ctx,
827     crypto_spi_ctx_template_t *template, crypto_mechanism_t *mechanism,
828     crypto_key_t *key, int kmflag)
829 {
830         int rv = CRYPTO_SUCCESS;

862 /* EXPORT DELETE START */

832         void *keysched;
833         size_t size;

835         if (template == NULL) {
836                 if ((keysched = blowfish_alloc_keysched(&size, kmflag)) == NULL)
837                         return (CRYPTO_HOST_MEMORY);
838                 /*
839                  * Initialize key schedule.
840                  * Key length is stored in the key.
841                  */
842                 if ((rv = init_keysched(key, keysched)) != CRYPTO_SUCCESS)
843                         kmem_free(keysched, size);

845                 blowfish_ctx->bc_flags |= PROVIDER_OWNS_KEY_SCHEDULE;
846                 blowfish_ctx->bc_keysched_len = size;
847         } else {
848                 keysched = template;
849         }
850         blowfish_ctx->bc_keysched = keysched;

852         switch (mechanism->cm_type) {
853         case BLOWFISH_CBC_MECH_INFO_TYPE:
854                 rv = cbc_init_ctx((cbc_ctx_t *)blowfish_ctx,
855                     mechanism->cm_param, mechanism->cm_param_len,
856                     BLOWFISH_BLOCK_LEN, blowfish_copy_block64);
857                 break;
858         case BLOWFISH_ECB_MECH_INFO_TYPE:
859                 blowfish_ctx->bc_flags |= ECB_MODE;
860         }

862         if (rv != CRYPTO_SUCCESS) {
863                 if (blowfish_ctx->bc_flags & PROVIDER_OWNS_KEY_SCHEDULE) {
864                         bzero(keysched, size);
865                         kmem_free(keysched, size);
866                 }
```



```
867         }

901 /* EXPORT DELETE END */

869         return (rv);
870 }
```

_____unchanged_portion_omitted_

```
**********************************************************
   40898 Thu Jul 11 01:29:52 2013
new/usr/src/uts/common/crypto/io/rsa.c
first pass
**********************************************************
______unchanged_portion_omitted_

 308 static int rsa_encrypt_common(rsa_mech_type_t, crypto_key_t *,
 309     crypto_data_t *, crypto_data_t *);
 310 static int rsa_decrypt_common(rsa_mech_type_t, crypto_key_t *,
 311     crypto_data_t *, crypto_data_t *);
 312 static int rsa_sign_common(rsa_mech_type_t, crypto_key_t *,
 313     crypto_data_t *, crypto_data_t *);
 314 static int rsa_verify_common(rsa_mech_type_t, crypto_key_t *,
 315     crypto_data_t *, crypto_data_t *);
 316 static int compare_data(crypto_data_t *, uchar_t *);

 318 /* EXPORT DELETE START */

 318 static int core_rsa_encrypt(crypto_key_t *, uchar_t *, int, uchar_t *, int);
 319 static int core_rsa_decrypt(crypto_key_t *, uchar_t *, int, uchar_t *);

 323 /* EXPORT DELETE END */

 321 static crypto_kcf_provider_handle_t rsa_prov_handle = NULL;

 323 int
 324 _init(void)
 325 {
 326         int ret;

 328         if ((ret = mod_install(&modlinkage)) != 0)
 329                 return (ret);

 331         /* Register with KCF.  If the registration fails, remove the module. */
 332         if (crypto_register_provider(&rsa_prov_info, &rsa_prov_handle)) {
 333                 (void) mod_remove(&modlinkage);
 334                 return (EACCES);
 335         }

 337         return (0);
 338 }
______unchanged_portion_omitted_

 367 static int
 368 check_mech_and_key(crypto_mechanism_t *mechanism, crypto_key_t *key)
 369 {
 370         int rv = CRYPTO_FAILED;

 376 /* EXPORT DELETE START */

 372         uchar_t *modulus;
 373         ssize_t modulus_len; /* In bytes */

 375         if (!RSA_VALID_MECH(mechanism))
 376                 return (CRYPTO_MECHANISM_INVALID);

 378         /*
 379          * We only support RSA keys that are passed as a list of
 380          * object attributes.
 381          */
 382         if (key->ck_format != CRYPTO_KEY_ATTR_LIST) {
 383                 return (CRYPTO_KEY_TYPE_INCONSISTENT);
 384         }

 386         if ((rv = crypto_get_key_attr(key, SUN_CKA_MODULUS, &modulus,
```



```
 387             &modulus_len)) != CRYPTO_SUCCESS) {
 388                 return (rv);
 389         }
 390         if (modulus_len < MIN_RSA_KEYLENGTH_IN_BYTES ||
 391             modulus_len > MAX_RSA_KEYLENGTH_IN_BYTES)
 392                 return (CRYPTO_KEY_SIZE_RANGE);

 400 /* EXPORT DELETE END */

 394         return (rv);
 395 }
______unchanged_portion_omitted_

 587 static int
 588 rsa_encrypt_common(rsa_mech_type_t mech_type, crypto_key_t *key,
 589     crypto_data_t *plaintext, crypto_data_t *ciphertext)
 590 {
 591         int rv = CRYPTO_FAILED;

 601 /* EXPORT DELETE START */

 593         int plen;
 594         uchar_t *ptptr;
 595         uchar_t *modulus;
 596         ssize_t modulus_len;
 597         uchar_t tmp_data[MAX_RSA_KEYLENGTH_IN_BYTES];
 598         uchar_t plain_data[MAX_RSA_KEYLENGTH_IN_BYTES];
 599         uchar_t cipher_data[MAX_RSA_KEYLENGTH_IN_BYTES];

 601         if ((rv = crypto_get_key_attr(key, SUN_CKA_MODULUS, &modulus,
 602             &modulus_len)) != CRYPTO_SUCCESS) {
 603                 return (rv);
 604         }

 606         plen = plaintext->cd_length;
 607         if (mech_type == RSA_PKCS_MECH_INFO_TYPE) {
 608                 if (plen > (modulus_len - MIN_PKCS1_PADLEN))
 609                         return (CRYPTO_DATA_LEN_RANGE);
 610         } else {
 611                 if (plen > modulus_len)
 612                         return (CRYPTO_DATA_LEN_RANGE);
 613         }

 615         /*
 616          * Output buf len must not be less than RSA modulus size.
 617          */
 618         if (ciphertext->cd_length < modulus_len) {
 619                 ciphertext->cd_length = modulus_len;
 620                 return (CRYPTO_BUFFER_TOO_SMALL);
 621         }

 623         ASSERT(plaintext->cd_length <= sizeof (tmp_data));
 624         if ((rv = crypto_get_input_data(plaintext, &ptptr, tmp_data))
 625             != CRYPTO_SUCCESS)
 626                 return (rv);

 628         if (mech_type == RSA_PKCS_MECH_INFO_TYPE) {
 629                 rv = pkcs1_encode(PKCS1_ENCRYPT, ptptr, plen,
 630                     plain_data, modulus_len);

 632                 if (rv != CRYPTO_SUCCESS)
 633                         return (rv);
 634         } else {
 635                 bzero(plain_data, modulus_len - plen);
 636                 bcopy(ptptr, &plain_data[modulus_len - plen], plen);
 637         }
```

```
639         rv = core_rsa_encrypt(key, plain_data, modulus_len, cipher_data, 1);
640         if (rv == CRYPTO_SUCCESS) {
641                 /* copy out to ciphertext */
642                 if ((rv = crypto_put_output_data(cipher_data,
643                     ciphertext, modulus_len)) != CRYPTO_SUCCESS)
644                         return (rv);

646                 ciphertext->cd_length = modulus_len;
647         }

659 /* EXPORT DELETE END */

649         return (rv);
650 }

664 /* EXPORT DELETE START */

652 static int
653 core_rsa_encrypt(crypto_key_t *key, uchar_t *in,
654     int in_len, uchar_t *out, int is_public)
655 {
656         int rv;
657         uchar_t *expo, *modulus;
658         ssize_t expo_len;
659         ssize_t modulus_len;
660         RSAbytekey k;

662         if (is_public) {
663                 if ((rv = crypto_get_key_attr(key, SUN_CKA_PUBLIC_EXPONENT,
664                     &expo, &expo_len)) != CRYPTO_SUCCESS)
665                         return (rv);
666         } else {
667                 /*
668                  * SUN_CKA_PRIVATE_EXPONENT is a required attribute for a
669                  * RSA secret key. See the comments in core_rsa_decrypt
670                  * routine which calls this routine with a private key.
671                  */
672                 if ((rv = crypto_get_key_attr(key, SUN_CKA_PRIVATE_EXPONENT,
673                     &expo, &expo_len)) != CRYPTO_SUCCESS)
674                         return (rv);
675         }

677         if ((rv = crypto_get_key_attr(key, SUN_CKA_MODULUS, &modulus,
678             &modulus_len)) != CRYPTO_SUCCESS) {
679                 return (rv);
680         }

682         k.modulus = modulus;
683         k.modulus_bits = CRYPTO_BYTES2BITS(modulus_len);
684         k.pubexpo = expo;
685         k.pubexpo_bytes = expo_len;
686         k.rfunc = NULL;

688         rv = rsa_encrypt(&k, in, in_len, out);

690         return (rv);
691 }

707 /* EXPORT DELETE END */

693 /* ARGSUSED */
694 static int
695 rsaprov_decrypt(crypto_ctx_t *ctx, crypto_data_t *ciphertext,
696     crypto_data_t *plaintext, crypto_req_handle_t req)
697 {
```



```
698         int rv;
699         rsa_ctx_t *ctxp;

701         ASSERT(ctx->cc_provider_private != NULL);
702         ctxp = ctx->cc_provider_private;

704         RSA_ARG_INPLACE(ciphertext, plaintext);

706         /* See the comments on KM_SLEEP flag in rsaprov_encrypt() */
707         rv = rsa_decrypt_common(ctxp->mech_type, ctxp->key,
708             ciphertext, plaintext);

710         if (rv != CRYPTO_BUFFER_TOO_SMALL)
711                 (void) rsa_free_context(ctx);

713         return (rv);
714 }
```

unchanged_portion_omitted_

```
733 static int
734 rsa_decrypt_common(rsa_mech_type_t mech_type, crypto_key_t *key,
735     crypto_data_t *ciphertext, crypto_data_t *plaintext)
736 {
737         int rv = CRYPTO_FAILED;

755 /* EXPORT DELETE START */

739         size_t plain_len;
740         uchar_t *ctptr;
741         uchar_t *modulus;
742         ssize_t modulus_len;
743         uchar_t plain_data[MAX_RSA_KEYLENGTH_IN_BYTES];
744         uchar_t tmp_data[MAX_RSA_KEYLENGTH_IN_BYTES];

746         if ((rv = crypto_get_key_attr(key, SUN_CKA_MODULUS, &modulus,
747             &modulus_len)) != CRYPTO_SUCCESS) {
748                 return (rv);
749         }

751         /*
752          * Ciphertext length must be equal to RSA modulus size.
753          */
754         if (ciphertext->cd_length != modulus_len)
755                 return (CRYPTO_ENCRYPTED_DATA_LEN_RANGE);

757         ASSERT(ciphertext->cd_length <= sizeof (tmp_data));
758         if ((rv = crypto_get_input_data(ciphertext, &ctptr, tmp_data))
759             != CRYPTO_SUCCESS)
760                 return (rv);

762         rv = core_rsa_decrypt(key, ctptr, modulus_len, plain_data);
763         if (rv == CRYPTO_SUCCESS) {
764                 plain_len = modulus_len;

766                 if (mech_type == RSA_PKCS_MECH_INFO_TYPE) {
767                         /* Strip off the PKCS block formatting data. */
768                         rv = pkcs1_decode(PKCS1_DECRYPT, plain_data,
769                             &plain_len);
770                         if (rv != CRYPTO_SUCCESS)
771                                 return (rv);
772                 }

774                 if (plain_len > plaintext->cd_length) {
775                         plaintext->cd_length = plain_len;
776                         return (CRYPTO_BUFFER_TOO_SMALL);
777                 }
```

```
779                 if ((rv = crypto_put_output_data(
780                     plain_data + modulus_len - plain_len,
781                     plaintext, plain_len)) != CRYPTO_SUCCESS)
782                         return (rv);

784                 plaintext->cd_length = plain_len;
785         }

805 /* EXPORT DELETE END */

787         return (rv);
788 }

810 /* EXPORT DELETE START */

790 static int
791 core_rsa_decrypt(crypto_key_t *key, uchar_t *in, int in_len, uchar_t *out)
792 {
793         int rv;
794         uchar_t *modulus, *prime1, *prime2, *expo1, *expo2, *coef;
795         ssize_t modulus_len;
796         ssize_t prime1_len, prime2_len;
797         ssize_t expo1_len, expo2_len, coef_len;
798         RSAbytekey k;

800         if ((rv = crypto_get_key_attr(key, SUN_CKA_MODULUS, &modulus,
801             &modulus_len)) != CRYPTO_SUCCESS) {
802                 return (rv);
803         }

805         /*
806          * The following attributes are not required to be
807          * present in a RSA secret key. If any of them is not present
808          * we call the encrypt routine with a flag indicating use of
809          * private exponent (d). Note that SUN_CKA_PRIVATE_EXPONENT is
810          * a required attribute for a RSA secret key.
811          */
812         if ((crypto_get_key_attr(key, SUN_CKA_PRIME_1, &prime1, &prime1_len)
813             != CRYPTO_SUCCESS) ||
814             (crypto_get_key_attr(key, SUN_CKA_PRIME_2, &prime2, &prime2_len)
815             != CRYPTO_SUCCESS) ||
816             (crypto_get_key_attr(key, SUN_CKA_EXPONENT_1, &expo1, &expo1_len)
817             != CRYPTO_SUCCESS) ||
818             (crypto_get_key_attr(key, SUN_CKA_EXPONENT_2, &expo2, &expo2_len)
819             != CRYPTO_SUCCESS) ||
820             (crypto_get_key_attr(key, SUN_CKA_COEFFICIENT, &coef, &coef_len)
821             != CRYPTO_SUCCESS)) {
822                 return (core_rsa_encrypt(key, in, in_len, out, 0));
823         }

825         k.modulus = modulus;
826         k.modulus_bits = CRYPTO_BYTES2BITS(modulus_len);
827         k.prime1 = prime1;
828         k.prime1_bytes = prime1_len;
829         k.prime2 = prime2;
830         k.prime2_bytes = prime2_len;
831         k.expo1 = expo1;
832         k.expo1_bytes = expo1_len;
833         k.expo2 = expo2;
834         k.expo2_bytes = expo2_len;
835         k.coeff = coef;
836         k.coeff_bytes = coef_len;
837         k.rfunc = NULL;

839         rv = rsa_decrypt(&k, in, in_len, out);
```



```
841         return (rv);
842 }

866 /* EXPORT DELETE END */

844 /* ARGSUSED */
845 static int
846 rsa_sign_verify_common_init(crypto_ctx_t *ctx, crypto_mechanism_t *mechanism,
847     crypto_key_t *key, crypto_spi_ctx_template_t ctx_template,
848     crypto_req_handle_t req)
849 {
850         int rv;
851         int kmflag;
852         rsa_ctx_t *ctxp;
853         digest_rsa_ctx_t *dctxp;

855         if ((rv = check_mech_and_key(mechanism, key)) != CRYPTO_SUCCESS)
856                 return (rv);

858         /*
859          * Allocate a RSA context.
860          */
861         kmflag = crypto_kmflag(req);
862         switch (mechanism->cm_type) {
863         case MD5_RSA_PKCS_MECH_INFO_TYPE:
864         case SHA1_RSA_PKCS_MECH_INFO_TYPE:
865         case SHA256_RSA_PKCS_MECH_INFO_TYPE:
866         case SHA384_RSA_PKCS_MECH_INFO_TYPE:
867         case SHA512_RSA_PKCS_MECH_INFO_TYPE:
868                 dctxp = kmem_zalloc(sizeof (digest_rsa_ctx_t), kmflag);
869                 ctxp = (rsa_ctx_t *)dctxp;
870                 break;
871         default:
872                 ctxp = kmem_zalloc(sizeof (rsa_ctx_t), kmflag);
873                 break;
874         }

876         if (ctxp == NULL)
877                 return (CRYPTO_HOST_MEMORY);

879         ctxp->mech_type = mechanism->cm_type;
880         if ((rv = crypto_copy_key_to_ctx(key, &ctxp->key, &ctxp->keychunk_size,
881             kmflag)) != CRYPTO_SUCCESS) {
882                 switch (mechanism->cm_type) {
883                 case MD5_RSA_PKCS_MECH_INFO_TYPE:
884                 case SHA1_RSA_PKCS_MECH_INFO_TYPE:
885                 case SHA256_RSA_PKCS_MECH_INFO_TYPE:
886                 case SHA384_RSA_PKCS_MECH_INFO_TYPE:
887                 case SHA512_RSA_PKCS_MECH_INFO_TYPE:
888                         kmem_free(dctxp, sizeof (digest_rsa_ctx_t));
889                         break;
890                 default:
891                         kmem_free(ctxp, sizeof (rsa_ctx_t));
892                         break;
893                 }
894                 return (rv);
895         }

897         switch (mechanism->cm_type) {
898         case MD5_RSA_PKCS_MECH_INFO_TYPE:
899                 MD5Init(&(dctxp->md5_ctx));
900                 break;

902         case SHA1_RSA_PKCS_MECH_INFO_TYPE:
903                 SHA1Init(&(dctxp->sha1_ctx));
```

```
904                 break;

906         case SHA256_RSA_PKCS_MECH_INFO_TYPE:
907                 SHA2Init(SHA256, &(dctxp->sha2_ctx));
908                 break;

910         case SHA384_RSA_PKCS_MECH_INFO_TYPE:
911                 SHA2Init(SHA384, &(dctxp->sha2_ctx));
912                 break;

914         case SHA512_RSA_PKCS_MECH_INFO_TYPE:
915                 SHA2Init(SHA512, &(dctxp->sha2_ctx));
916                 break;
917         }

919         ctx->cc_provider_private = ctxp;

921         return (CRYPTO_SUCCESS);
922 }

924 #define SHA1_DIGEST_SIZE 20
925 #define MD5_DIGEST_SIZE 16

927 #define INIT_RAW_CRYPTO_DATA(data, base, len, cd_len)   \
928         (data).cd_format = CRYPTO_DATA_RAW;              \
929         (data).cd_offset = 0;                            \
930         (data).cd_raw.iov_base = (char *)base;           \
931         (data).cd_raw.iov_len = len;                     \
932         (data).cd_length = cd_len;

934 static int
935 rsa_digest_svrfy_common(digest_rsa_ctx_t *ctxp, crypto_data_t *data,
936     crypto_data_t *signature, uchar_t flag)
937 {
938         int rv = CRYPTO_FAILED;

964 /* EXPORT DELETE START */

940         uchar_t digest[SHA512_DIGEST_LENGTH];
941         /* The der_data size is enough for MD5 also */
942         uchar_t der_data[SHA512_DIGEST_LENGTH + SHA2_DER_PREFIX_Len];
943         ulong_t der_data_len;
944         crypto_data_t der_cd;
945         rsa_mech_type_t mech_type;

947         ASSERT(flag & CRYPTO_DO_SIGN || flag & CRYPTO_DO_VERIFY);
948         ASSERT(data != NULL || (flag & CRYPTO_DO_FINAL));

950         mech_type = ctxp->mech_type;
951         if (mech_type == RSA_PKCS_MECH_INFO_TYPE ||
952             mech_type == RSA_X_509_MECH_INFO_TYPE)
953                 return (CRYPTO_MECHANISM_INVALID);

955         /*
956          * We need to do the BUFFER_TOO_SMALL check before digesting
957          * the data. No check is needed for verify as signature is not
958          * an output argument for verify.
959          */
960         if (flag & CRYPTO_DO_SIGN) {
961                 uchar_t *modulus;
962                 ssize_t modulus_len;

964                 if ((rv = crypto_get_key_attr(ctxp->key, SUN_CKA_MODULUS,
965                     &modulus, &modulus_len)) != CRYPTO_SUCCESS) {
966                         return (rv);
967                 }
```



```
969                 if (signature->cd_length < modulus_len) {
970                         signature->cd_length = modulus_len;
971                         return (CRYPTO_BUFFER_TOO_SMALL);
972                 }
973         }

975         if (mech_type == MD5_RSA_PKCS_MECH_INFO_TYPE)
976                 rv = crypto_digest_data(data, &(ctxp->md5_ctx),
977                     digest, MD5Update, MD5Final, flag | CRYPTO_DO_MD5);

979         else if (mech_type == SHA1_RSA_PKCS_MECH_INFO_TYPE)
980                 rv = crypto_digest_data(data, &(ctxp->sha1_ctx),
981                     digest, SHA1Update, SHA1Final,  flag | CRYPTO_DO_SHA1);

983         else
984                 rv = crypto_digest_data(data, &(ctxp->sha2_ctx),
985                     digest, SHA2Update, SHA2Final, flag | CRYPTO_DO_SHA2);

987         if (rv != CRYPTO_SUCCESS)
988                 return (rv);


991         /*
992          * Prepare the DER encoding of the DigestInfo value as follows:
993          * MD5:         MD5_DER_PREFIX || H
994          * SHA-1:       SHA1_DER_PREFIX || H
995          *
996          * See rsa_impl.c for more details.
997          */
998         switch (mech_type) {
999         case MD5_RSA_PKCS_MECH_INFO_TYPE:
1000                 bcopy(MD5_DER_PREFIX, der_data, MD5_DER_PREFIX_Len);
1001                 bcopy(digest, der_data + MD5_DER_PREFIX_Len, MD5_DIGEST_SIZE);
1002                 der_data_len = MD5_DER_PREFIX_Len + MD5_DIGEST_SIZE;
1003                 break;

1005        case SHA1_RSA_PKCS_MECH_INFO_TYPE:
1006                bcopy(SHA1_DER_PREFIX, der_data, SHA1_DER_PREFIX_Len);
1007                bcopy(digest, der_data + SHA1_DER_PREFIX_Len,
1008                    SHA1_DIGEST_SIZE);
1009                der_data_len = SHA1_DER_PREFIX_Len + SHA1_DIGEST_SIZE;
1010                break;

1012        case SHA256_RSA_PKCS_MECH_INFO_TYPE:
1013                bcopy(SHA256_DER_PREFIX, der_data, SHA2_DER_PREFIX_Len);
1014                bcopy(digest, der_data + SHA2_DER_PREFIX_Len,
1015                    SHA256_DIGEST_LENGTH);
1016                der_data_len = SHA2_DER_PREFIX_Len + SHA256_DIGEST_LENGTH;
1017                break;

1019        case SHA384_RSA_PKCS_MECH_INFO_TYPE:
1020                bcopy(SHA384_DER_PREFIX, der_data, SHA2_DER_PREFIX_Len);
1021                bcopy(digest, der_data + SHA2_DER_PREFIX_Len,
1022                    SHA384_DIGEST_LENGTH);
1023                der_data_len = SHA2_DER_PREFIX_Len + SHA384_DIGEST_LENGTH;
1024                break;

1026        case SHA512_RSA_PKCS_MECH_INFO_TYPE:
1027                bcopy(SHA512_DER_PREFIX, der_data, SHA2_DER_PREFIX_Len);
1028                bcopy(digest, der_data + SHA2_DER_PREFIX_Len,
1029                    SHA512_DIGEST_LENGTH);
1030                der_data_len = SHA2_DER_PREFIX_Len + SHA512_DIGEST_LENGTH;
1031                break;
1032        }
```

```
1034         INIT_RAW_CRYPTO_DATA(der_cd, der_data, der_data_len, der_data_len);
1035         /*
1036          * Now, we are ready to sign or verify the DER_ENCODED data.
1037          */
1038         if (flag & CRYPTO_DO_SIGN)
1039                 rv = rsa_sign_common(mech_type, ctxp->key, &der_cd,
1040                     signature);
1041         else
1042                 rv = rsa_verify_common(mech_type, ctxp->key, &der_cd,
1043                     signature);

1071 /* EXPORT DELETE END */

1045         return (rv);
1046 }

1048 static int
1049 rsa_sign_common(rsa_mech_type_t mech_type, crypto_key_t *key,
1050     crypto_data_t *data, crypto_data_t *signature)
1051 {
1052         int rv = CRYPTO_FAILED;

1082 /* EXPORT DELETE START */

1054         int dlen;
1055         uchar_t *dataptr, *modulus;
1056         ssize_t modulus_len;
1057         uchar_t tmp_data[MAX_RSA_KEYLENGTH_IN_BYTES];
1058         uchar_t plain_data[MAX_RSA_KEYLENGTH_IN_BYTES];
1059         uchar_t signed_data[MAX_RSA_KEYLENGTH_IN_BYTES];

1061         if ((rv = crypto_get_key_attr(key, SUN_CKA_MODULUS, &modulus,
1062             &modulus_len)) != CRYPTO_SUCCESS) {
1063                 return (rv);
1064         }

1066         dlen = data->cd_length;
1067         switch (mech_type) {
1068         case RSA_PKCS_MECH_INFO_TYPE:
1069                 if (dlen > (modulus_len - MIN_PKCS1_PADLEN))
1070                         return (CRYPTO_DATA_LEN_RANGE);
1071                 break;
1072         case RSA_X_509_MECH_INFO_TYPE:
1073                 if (dlen > modulus_len)
1074                         return (CRYPTO_DATA_LEN_RANGE);
1075                 break;
1076         }

1078         if (signature->cd_length < modulus_len) {
1079                 signature->cd_length = modulus_len;
1080                 return (CRYPTO_BUFFER_TOO_SMALL);
1081         }

1083         ASSERT(data->cd_length <= sizeof (tmp_data));
1084         if ((rv = crypto_get_input_data(data, &dataptr, tmp_data))
1085             != CRYPTO_SUCCESS)
1086                 return (rv);

1088         switch (mech_type) {
1089         case RSA_PKCS_MECH_INFO_TYPE:
1090         case MD5_RSA_PKCS_MECH_INFO_TYPE:
1091         case SHA1_RSA_PKCS_MECH_INFO_TYPE:
1092         case SHA256_RSA_PKCS_MECH_INFO_TYPE:
1093         case SHA384_RSA_PKCS_MECH_INFO_TYPE:
1094         case SHA512_RSA_PKCS_MECH_INFO_TYPE:
1095                 /*
```



```
1096                  * Add PKCS padding to the input data to format a block
1097                  * type "01" encryption block.
1098                  */
1099                 rv = pkcs1_encode(PKCS1_SIGN, dataptr, dlen, plain_data,
1100                     modulus_len);
1101                 if (rv != CRYPTO_SUCCESS)
1102                         return (rv);

1104                 break;

1106         case RSA_X_509_MECH_INFO_TYPE:
1107                 bzero(plain_data, modulus_len - dlen);
1108                 bcopy(dataptr, &plain_data[modulus_len - dlen], dlen);
1109                 break;
1110         }

1112         rv = core_rsa_decrypt(key, plain_data, modulus_len, signed_data);
1113         if (rv == CRYPTO_SUCCESS) {
1114                 /* copy out to signature */
1115                 if ((rv = crypto_put_output_data(signed_data,
1116                     signature, modulus_len)) != CRYPTO_SUCCESS)
1117                         return (rv);

1119                 signature->cd_length = modulus_len;
1120         }

1152 /* EXPORT DELETE END */

1122         return (rv);
1123 }
______unchanged_portion_omitted_

1262 static int
1263 rsa_verify_common(rsa_mech_type_t mech_type, crypto_key_t *key,
1264     crypto_data_t *data, crypto_data_t *signature)
1265 {
1266         int rv = CRYPTO_FAILED;

1300 /* EXPORT DELETE START */

1268         uchar_t *sigptr, *modulus;
1269         ssize_t modulus_len;
1270         uchar_t plain_data[MAX_RSA_KEYLENGTH_IN_BYTES];
1271         uchar_t tmp_data[MAX_RSA_KEYLENGTH_IN_BYTES];

1273         if ((rv = crypto_get_key_attr(key, SUN_CKA_MODULUS, &modulus,
1274             &modulus_len)) != CRYPTO_SUCCESS) {
1275                 return (rv);
1276         }

1278         if (signature->cd_length != modulus_len)
1279                 return (CRYPTO_SIGNATURE_LEN_RANGE);

1281         ASSERT(signature->cd_length <= sizeof (tmp_data));
1282         if ((rv = crypto_get_input_data(signature, &sigptr, tmp_data))
1283             != CRYPTO_SUCCESS)
1284                 return (rv);

1286         rv = core_rsa_encrypt(key, sigptr, modulus_len, plain_data, 1);
1287         if (rv != CRYPTO_SUCCESS)
1288                 return (rv);

1290         if (mech_type == RSA_X_509_MECH_INFO_TYPE) {
1291                 if (compare_data(data, (plain_data + modulus_len
1292                     - data->cd_length)) != 0)
1293                         rv = CRYPTO_SIGNATURE_INVALID;
```

```
1295         } else {
1296                 size_t data_len = modulus_len;

1298                 /*
1299                  * Strip off the encoded padding bytes in front of the
1300                  * recovered data, then compare the recovered data with
1301                  * the original data.
1302                  */
1303                 rv = pkcs1_decode(PKCS1_VERIFY, plain_data, &data_len);
1304                 if (rv != CRYPTO_SUCCESS)
1305                         return (rv);

1307                 if (data_len != data->cd_length)
1308                         return (CRYPTO_SIGNATURE_LEN_RANGE);

1310                 if (compare_data(data, (plain_data + modulus_len
1311                     - data_len)) != 0)
1312                         rv = CRYPTO_SIGNATURE_INVALID;
1313         }

1349 /* EXPORT DELETE END */

1315         return (rv);
1316 }
______unchanged_portion_omitted_

1464 static int
1465 rsa_verify_recover_common(rsa_mech_type_t mech_type, crypto_key_t *key,
1466     crypto_data_t *signature, crypto_data_t *data)
1467 {
1468         int rv = CRYPTO_FAILED;

1506 /* EXPORT DELETE START */

1470         size_t data_len;
1471         uchar_t *sigptr, *modulus;
1472         ssize_t modulus_len;
1473         uchar_t plain_data[MAX_RSA_KEYLENGTH_IN_BYTES];
1474         uchar_t tmp_data[MAX_RSA_KEYLENGTH_IN_BYTES];

1476         if ((rv = crypto_get_key_attr(key, SUN_CKA_MODULUS, &modulus,
1477             &modulus_len)) != CRYPTO_SUCCESS) {
1478                 return (rv);
1479         }

1481         if (signature->cd_length != modulus_len)
1482                 return (CRYPTO_SIGNATURE_LEN_RANGE);

1484         ASSERT(signature->cd_length <= sizeof (tmp_data));
1485         if ((rv = crypto_get_input_data(signature, &sigptr, tmp_data))
1486             != CRYPTO_SUCCESS)
1487                 return (rv);

1489         rv = core_rsa_encrypt(key, sigptr, modulus_len, plain_data, 1);
1490         if (rv != CRYPTO_SUCCESS)
1491                 return (rv);

1493         data_len = modulus_len;

1495         if (mech_type == RSA_PKCS_MECH_INFO_TYPE) {
1496                 /*
1497                  * Strip off the encoded padding bytes in front of the
1498                  * recovered data, then compare the recovered data with
1499                  * the original data.
1500                  */
```



```
1501                 rv = pkcs1_decode(PKCS1_VERIFY, plain_data, &data_len);
1502                 if (rv != CRYPTO_SUCCESS)
1503                         return (rv);
1504         }

1506         if (data->cd_length < data_len) {
1507                 data->cd_length = data_len;
1508                 return (CRYPTO_BUFFER_TOO_SMALL);
1509         }

1511         if ((rv = crypto_put_output_data(plain_data + modulus_len - data_len,
1512             data, data_len)) != CRYPTO_SUCCESS)
1513                 return (rv);
1514         data->cd_length = data_len;

1554 /* EXPORT DELETE END */

1516         return (rv);
1517 }
______unchanged_portion_omitted_
```

```
**********************************************************
    1322 Thu Jul 11 01:29:52 2013
new/usr/src/uts/common/des/Makefile
first pass
**********************************************************
#
# 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
#
# ident "%Z%%M% %I%     %E% SMI"
#
# Copyright (c) 1989,1999 by Sun Microsystems, Inc.
# All rights reserved.
#
# uts/common/des/Makefile
#
# include global definitions
include ../../../Makefile.master

HDRS=   des.h           desdata.h       softdes.h

ROOTDIRS= $(ROOT)/usr/include/des

ROOTHDRS= $(HDRS:%=$(ROOTDIRS)/%)

CHECKHDRS= $(HDRS:%.h=%.check)

# install rule
$(ROOTDIRS)/%: %
        $(INS.file)

.KEEP_STATE:

.PARALLEL: $(CHECKHDRS)

install_h: $(ROOTDIRS) $(ROOTHDRS)

$(ROOTDIRS):
        $(INS.dir)

# EXPORT DELETE START
EXPORT_SRC:
        $(RM) Makefile+ des_crypt.c+ des_soft.c+ desdata.h+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < des_crypt.c > des_crypt.c+
        $(MV) des_crypt.c+ des_crypt.c
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < des_soft.c > des_soft.c+
        $(MV) des_soft.c+ des_soft.c
```



```
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < desdata.h > desdata.h+
        $(MV) desdata.h+ desdata.h
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(RM) Makefile
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile des_crypt.c des_soft.c desdata.h

# EXPORT DELETE END
check:  $(CHECKHDRS)
```

```
**********************************************************
   28843 Thu Jul 11 01:29:53 2013
new/usr/src/uts/common/des/des_crypt.c
first pass
**********************************************************
```

```
/*
 * 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 2010 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*      Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989 AT&T     */
/*        All Rights Reserved   */

/*
 * Portions of this source code were derived from Berkeley 4.3 BSD
 * under license from the Regents of the University of California.
 */

/*
 * des_crypt.c, DES encryption library routines
 */

#include <sys/errno.h>
#include <sys/modctl.h>

#include <sys/systm.h>
#include <sys/cmn_err.h>
#include <sys/ddi.h>
#include <sys/crypto/common.h>
#include <sys/crypto/spi.h>
#include <sys/sysmacros.h>
#include <sys/strsun.h>
#include <sys/note.h>
#include <modes/modes.h>
#define _DES_IMPL
#include <des/des_impl.h>

/* EXPORT DELETE START */
#include <sys/types.h>
#include <rpc/des_crypt.h>
#include <des/des.h>

#ifdef sun_hardware
#include <sys/ioctl.h>
#ifdef _KERNEL
```



```
#include <sys/conf.h>
static int g_desfd = -1;
#define getdesfd()      (cdevsw[11].d_open(0, 0) ? -1 : 0)
#define ioctl(a, b, c)  (cdevsw[11].d_ioctl(0, b, c, 0) ? -1 : 0)
#else
#define getdesfd()      (open("/dev/des", 0, 0))
#endif  /* _KERNEL */
#endif  /* sun */

static int common_crypt(char *key, char *buf, size_t len,
    unsigned int mode, struct desparams *desp);

extern int _des_crypt(char *buf, size_t len, struct desparams *desp);

/* EXPORT DELETE END */

extern struct mod_ops mod_cryptoops;

/*
 * Module linkage information for the kernel.
 */
static struct modlmisc modlmisc = {
        &mod_miscops,
        "des encryption",
};
```

_____*unchanged_portion_omitted_*

```
/* EXPORT DELETE START */

#define DES_MIN_KEY_LEN         DES_MINBYTES
#define DES_MAX_KEY_LEN         DES_MAXBYTES
#define DES3_MIN_KEY_LEN        DES3_MAXBYTES   /* no CKK_DES2 support */
#define DES3_MAX_KEY_LEN        DES3_MAXBYTES

/* EXPORT DELETE END */

#ifndef DES_MIN_KEY_LEN
#define DES_MIN_KEY_LEN         0
#endif

#ifndef DES_MAX_KEY_LEN
#define DES_MAX_KEY_LEN         0
#endif

#ifndef DES3_MIN_KEY_LEN
#define DES3_MIN_KEY_LEN        0
#endif

#ifndef DES3_MAX_KEY_LEN
#define DES3_MAX_KEY_LEN        0
#endif


/*
 * Mechanism info structure passed to KCF during registration.
 */
static crypto_mech_info_t des_mech_info_tab[] = {
        /* DES_ECB */
        {SUN_CKM_DES_ECB, DES_ECB_MECH_INFO_TYPE,
            CRYPTO_FG_ENCRYPT | CRYPTO_FG_ENCRYPT_ATOMIC |
            CRYPTO_FG_DECRYPT | CRYPTO_FG_DECRYPT_ATOMIC,
            DES_MIN_KEY_LEN, DES_MAX_KEY_LEN, CRYPTO_KEYSIZE_UNIT_IN_BYTES},
        /* DES_CBC */
        {SUN_CKM_DES_CBC, DES_CBC_MECH_INFO_TYPE,
            CRYPTO_FG_ENCRYPT | CRYPTO_FG_ENCRYPT_ATOMIC |
            CRYPTO_FG_DECRYPT | CRYPTO_FG_DECRYPT_ATOMIC,
```

```
132             DES_MIN_KEY_LEN, DES_MAX_KEY_LEN, CRYPTO_KEYSIZE_UNIT_IN_BYTES},
133         /* DES3_ECB */
134         {SUN_CKM_DES3_ECB, DES3_ECB_MECH_INFO_TYPE,
135             CRYPTO_FG_ENCRYPT | CRYPTO_FG_ENCRYPT_ATOMIC |
136             CRYPTO_FG_DECRYPT | CRYPTO_FG_DECRYPT_ATOMIC,
137             DES3_MIN_KEY_LEN, DES3_MAX_KEY_LEN, CRYPTO_KEYSIZE_UNIT_IN_BYTES},
138         /* DES3_CBC */
139         {SUN_CKM_DES3_CBC, DES3_CBC_MECH_INFO_TYPE,
140             CRYPTO_FG_ENCRYPT | CRYPTO_FG_ENCRYPT_ATOMIC |
141             CRYPTO_FG_DECRYPT | CRYPTO_FG_DECRYPT_ATOMIC,
142             DES3_MIN_KEY_LEN, DES3_MAX_KEY_LEN, CRYPTO_KEYSIZE_UNIT_IN_BYTES}
143 };
______unchanged_portion_omitted_

299 /*
300  * CBC mode encryption
301  */
302 /* ARGSUSED */
303 int
304 cbc_crypt(char *key, char *buf, size_t len, unsigned int mode, char *ivec)
305 {
306         int err = 0;
314 /* EXPORT DELETE START */
307         struct desparams dp;

309         dp.des_mode = CBC;
310         COPY8(ivec, dp.des_ivec);
311         err = common_crypt(key, buf, len, mode, &dp);
312         COPY8(dp.des_ivec, ivec);
321 /* EXPORT DELETE END */
313         return (err);
314 }


317 /*
318  * ECB mode encryption
319  */
320 /* ARGSUSED */
321 int
322 ecb_crypt(char *key, char *buf, size_t len, unsigned int mode)
323 {
324         int err = 0;
334 /* EXPORT DELETE START */
325         struct desparams dp;

327         dp.des_mode = ECB;
328         err = common_crypt(key, buf, len, mode, &dp);
339 /* EXPORT DELETE END */
329         return (err);
330 }



345 /* EXPORT DELETE START */
334 /*
335  * Common code to cbc_crypt() & ecb_crypt()
336  */
337 static int
338 common_crypt(char *key, char *buf, size_t len, unsigned int mode,
339     struct desparams *desp)
340 {
341         int desdev;

343         if ((len % 8) != 0 || len > DES_MAXDATA)
344                 return (DESERR_BADPARAM);
```



```
346         desp->des_dir =
347             ((mode & DES_DIRMASK) == DES_ENCRYPT) ? ENCRYPT : DECRYPT;

349         desdev = mode & DES_DEVMASK;
350         COPY8(key, desp->des_key);

352 #ifdef sun_hardware
353         if (desdev == DES_HW) {
354                 int res;

356                 if (g_desfd < 0 &&
357                     (g_desfd == -1 || (g_desfd = getdesfd()) < 0))
358                                 goto software;  /* no hardware device */

360                 /*
361                  * hardware
362                  */
363                 desp->des_len = len;
364                 if (len <= DES_QUICKLEN) {
365                         DESCOPY(buf, desp->des_data, len);
366                         res = ioctl(g_desfd, DESIOCQUICK, (char *)desp);
367                         DESCOPY(desp->des_data, buf, len);
368                 } else {
369                         desp->des_buf = (uchar_t *)buf;
370                         res = ioctl(g_desfd, DESIOCBLOCK, (char *)desp);
371                 }
372                 return (res == 0 ? DESERR_NONE : DESERR_HWERROR);
373         }
374 software:
375 #endif
376         /*
377          * software
378          */
379         if (!_des_crypt(buf, len, desp))
380                 return (DESERR_HWERROR);

382         return (desdev == DES_SW ? DESERR_NONE : DESERR_NOHWDEVICE);
383 }
______unchanged_portion_omitted_

431 /* EXPORT DELETE END */

419 /*
420  * KCF software provider control entry points.
421  */
422 /* ARGSUSED */
423 static void
424 des_provider_status(crypto_provider_handle_t provider, uint_t *status)
425 {
426         *status = CRYPTO_PROVIDER_READY;
427 }

429 /*
430  * KCF software provider encrypt entry points.
431  */
432 static int
433 des_common_init(crypto_ctx_t *ctx, crypto_mechanism_t *mechanism,
434     crypto_key_t *key, crypto_spi_ctx_template_t template,
435     crypto_req_handle_t req)
436 {

452 /* EXPORT DELETE START */

438         des_strength_t strength;
439         des_ctx_t *des_ctx = NULL;
440         int rv;
```

```
441         int kmflag;

443         /*
444          * Only keys by value are supported by this module.
445          */
446         if (key->ck_format != CRYPTO_KEY_RAW) {
447                 return (CRYPTO_KEY_TYPE_INCONSISTENT);
448         }

450         kmflag = crypto_kmflag(req);
451         /* Check mechanism type and parameter length */
452         switch (mechanism->cm_type) {
453         case DES_ECB_MECH_INFO_TYPE:
454                 des_ctx = ecb_alloc_ctx(kmflag);
455                 /* FALLTHRU */
456         case DES_CBC_MECH_INFO_TYPE:
457                 if (mechanism->cm_param != NULL &&
458                     mechanism->cm_param_len != DES_BLOCK_LEN)
459                         return (CRYPTO_MECHANISM_PARAM_INVALID);
460                 if (key->ck_length != DES_MAXBITS)
461                         return (CRYPTO_KEY_SIZE_RANGE);
462                 strength = DES;
463                 if (des_ctx == NULL)
464                         des_ctx = cbc_alloc_ctx(kmflag);
465                 break;
466         case DES3_ECB_MECH_INFO_TYPE:
467                 des_ctx = ecb_alloc_ctx(kmflag);
468                 /* FALLTHRU */
469         case DES3_CBC_MECH_INFO_TYPE:
470                 if (mechanism->cm_param != NULL &&
471                     mechanism->cm_param_len != DES_BLOCK_LEN)
472                         return (CRYPTO_MECHANISM_PARAM_INVALID);
473                 if (key->ck_length != DES3_MAXBITS)
474                         return (CRYPTO_KEY_SIZE_RANGE);
475                 strength = DES3;
476                 if (des_ctx == NULL)
477                         des_ctx = cbc_alloc_ctx(kmflag);
478                 break;
479         default:
480                 return (CRYPTO_MECHANISM_INVALID);
481         }

483         if ((rv = des_common_init_ctx(des_ctx, template, mechanism, key,
484             strength, kmflag)) != CRYPTO_SUCCESS) {
485                 crypto_free_mode_ctx(des_ctx);
486                 return (rv);
487         }

489         ctx->cc_provider_private = des_ctx;

507 /* EXPORT DELETE END */

491         return (CRYPTO_SUCCESS);
492 }
_____unchanged_portion_omitted_

527 /* ARGSUSED */
528 static int
529 des_encrypt(crypto_ctx_t *ctx, crypto_data_t *plaintext,
530     crypto_data_t *ciphertext, crypto_req_handle_t req)
531 {
532         int ret;

552 /* EXPORT DELETE START */
534         des_ctx_t *des_ctx;
```



```
536         /*
537          * Plaintext must be a multiple of the block size.
538          * This test only works for non-padded mechanisms
539          * when blocksize is 2^N.
540          */
541         if ((plaintext->cd_length & (DES_BLOCK_LEN - 1)) != 0)
542                 return (CRYPTO_DATA_LEN_RANGE);

544         ASSERT(ctx->cc_provider_private != NULL);
545         des_ctx = ctx->cc_provider_private;

547         DES_ARG_INPLACE(plaintext, ciphertext);

549         /*
550          * We need to just return the length needed to store the output.
551          * We should not destroy the context for the following case.
552          */
553         if (ciphertext->cd_length < plaintext->cd_length) {
554                 ciphertext->cd_length = plaintext->cd_length;
555                 return (CRYPTO_BUFFER_TOO_SMALL);
556         }

558         /*
559          * Do an update on the specified input data.
560          */
561         ret = des_encrypt_update(ctx, plaintext, ciphertext, req);
562         ASSERT(des_ctx->dc_remainder_len == 0);
563         (void) des_free_context(ctx);

584 /* EXPORT DELETE END */

565         /* LINTED */
566         return (ret);
567 }

569 /* ARGSUSED */
570 static int
571 des_decrypt(crypto_ctx_t *ctx, crypto_data_t *ciphertext,
572     crypto_data_t *plaintext, crypto_req_handle_t req)
573 {
574         int ret;

597 /* EXPORT DELETE START */
576         des_ctx_t *des_ctx;

578         /*
579          * Ciphertext must be a multiple of the block size.
580          * This test only works for non-padded mechanisms
581          * when blocksize is 2^N.
582          */
583         if ((ciphertext->cd_length & (DES_BLOCK_LEN - 1)) != 0)
584                 return (CRYPTO_ENCRYPTED_DATA_LEN_RANGE);

586         ASSERT(ctx->cc_provider_private != NULL);
587         des_ctx = ctx->cc_provider_private;

589         DES_ARG_INPLACE(ciphertext, plaintext);

591         /*
592          * We need to just return the length needed to store the output.
593          * We should not destroy the context for the following case.
594          */
595         if (plaintext->cd_length < ciphertext->cd_length) {
596                 plaintext->cd_length = ciphertext->cd_length;
597                 return (CRYPTO_BUFFER_TOO_SMALL);
598         }
```

```
600         /*
601          * Do an update on the specified input data.
602          */
603         ret = des_decrypt_update(ctx, ciphertext, plaintext, req);
604         ASSERT(des_ctx->dc_remainder_len == 0);
605         (void) des_free_context(ctx);

629 /* EXPORT DELETE END */

607         /* LINTED */
608         return (ret);
609 }

611 /* ARGSUSED */
612 static int
613 des_encrypt_update(crypto_ctx_t *ctx, crypto_data_t *plaintext,
614     crypto_data_t *ciphertext, crypto_req_handle_t req)
615 {
616         off_t saved_offset;
617         size_t saved_length, out_len;
618         int ret = CRYPTO_SUCCESS;

644 /* EXPORT DELETE START */

620         ASSERT(ctx->cc_provider_private != NULL);

622         DES_ARG_INPLACE(plaintext, ciphertext);

624         /* compute number of bytes that will hold the ciphertext */
625         out_len = ((des_ctx_t *)ctx->cc_provider_private)->dc_remainder_len;
626         out_len += plaintext->cd_length;
627         out_len &= ~(DES_BLOCK_LEN - 1);

629         /* return length needed to store the output */
630         if (ciphertext->cd_length < out_len) {
631                 ciphertext->cd_length = out_len;
632                 return (CRYPTO_BUFFER_TOO_SMALL);
633         }

635         saved_offset = ciphertext->cd_offset;
636         saved_length = ciphertext->cd_length;

638         /*
639          * Do the DES update on the specified input data.
640          */
641         switch (plaintext->cd_format) {
642         case CRYPTO_DATA_RAW:
643                 ret = crypto_update_iov(ctx->cc_provider_private,
644                     plaintext, ciphertext, des_encrypt_contiguous_blocks,
645                     des_copy_block64);
646                 break;
647         case CRYPTO_DATA_UIO:
648                 ret = crypto_update_uio(ctx->cc_provider_private,
649                     plaintext, ciphertext, des_encrypt_contiguous_blocks,
650                     des_copy_block64);
651                 break;
652         case CRYPTO_DATA_MBLK:
653                 ret = crypto_update_mp(ctx->cc_provider_private,
654                     plaintext, ciphertext, des_encrypt_contiguous_blocks,
655                     des_copy_block64);
656                 break;
657         default:
658                 ret = CRYPTO_ARGUMENTS_BAD;
659         }
```



```
661         if (ret == CRYPTO_SUCCESS) {
662                 if (plaintext != ciphertext)
663                         ciphertext->cd_length =
664                             ciphertext->cd_offset - saved_offset;
665         } else {
666                 ciphertext->cd_length = saved_length;
667         }
668         ciphertext->cd_offset = saved_offset;

696 /* EXPORT DELETE END */

670         return (ret);
671 }

673 /* ARGSUSED */
674 static int
675 des_decrypt_update(crypto_ctx_t *ctx, crypto_data_t *ciphertext,
676     crypto_data_t *plaintext, crypto_req_handle_t req)
677 {
678         off_t saved_offset;
679         size_t saved_length, out_len;
680         int ret = CRYPTO_SUCCESS;

710 /* EXPORT DELETE START */

682         ASSERT(ctx->cc_provider_private != NULL);

684         DES_ARG_INPLACE(ciphertext, plaintext);

686         /* compute number of bytes that will hold the plaintext */
687         out_len = ((des_ctx_t *)ctx->cc_provider_private)->dc_remainder_len;
688         out_len += ciphertext->cd_length;
689         out_len &= ~(DES_BLOCK_LEN - 1);

691         /* return length needed to store the output */
692         if (plaintext->cd_length < out_len) {
693                 plaintext->cd_length = out_len;
694                 return (CRYPTO_BUFFER_TOO_SMALL);
695         }

697         saved_offset = plaintext->cd_offset;
698         saved_length = plaintext->cd_length;

700         /*
701          * Do the DES update on the specified input data.
702          */
703         switch (ciphertext->cd_format) {
704         case CRYPTO_DATA_RAW:
705                 ret = crypto_update_iov(ctx->cc_provider_private,
706                     ciphertext, plaintext, des_decrypt_contiguous_blocks,
707                     des_copy_block64);
708                 break;
709         case CRYPTO_DATA_UIO:
710                 ret = crypto_update_uio(ctx->cc_provider_private,
711                     ciphertext, plaintext, des_decrypt_contiguous_blocks,
712                     des_copy_block64);
713                 break;
714         case CRYPTO_DATA_MBLK:
715                 ret = crypto_update_mp(ctx->cc_provider_private,
716                     ciphertext, plaintext, des_decrypt_contiguous_blocks,
717                     des_copy_block64);
718                 break;
719         default:
720                 ret = CRYPTO_ARGUMENTS_BAD;
721         }
```

```
723         if (ret == CRYPTO_SUCCESS) {
724                 if (ciphertext != plaintext)
725                         plaintext->cd_length =
726                             plaintext->cd_offset - saved_offset;
727         } else {
728                 plaintext->cd_length = saved_length;
729         }
730         plaintext->cd_offset = saved_offset;

762 /* EXPORT DELETE END */

732         return (ret);
733 }

735 /* ARGSUSED */
736 static int
737 des_encrypt_final(crypto_ctx_t *ctx, crypto_data_t *ciphertext,
738     crypto_req_handle_t req)
739 {

773 /* EXPORT DELETE START */

740         des_ctx_t *des_ctx;

742         ASSERT(ctx->cc_provider_private != NULL);
743         des_ctx = ctx->cc_provider_private;

745         /*
746          * There must be no unprocessed plaintext.
747          * This happens if the length of the last data is
748          * not a multiple of the DES block length.
749          */
750         if (des_ctx->dc_remainder_len > 0)
751                 return (CRYPTO_DATA_LEN_RANGE);

753         (void) des_free_context(ctx);
754         ciphertext->cd_length = 0;

791 /* EXPORT DELETE END */

756         return (CRYPTO_SUCCESS);
757 }

759 /* ARGSUSED */
760 static int
761 des_decrypt_final(crypto_ctx_t *ctx, crypto_data_t *plaintext,
762     crypto_req_handle_t req)
763 {

802 /* EXPORT DELETE START */

764         des_ctx_t *des_ctx;

766         ASSERT(ctx->cc_provider_private != NULL);
767         des_ctx = ctx->cc_provider_private;

769         /*
770          * There must be no unprocessed ciphertext.
771          * This happens if the length of the last ciphertext is
772          * not a multiple of the DES block length.
773          */
774         if (des_ctx->dc_remainder_len > 0)
775                 return (CRYPTO_ENCRYPTED_DATA_LEN_RANGE);

777         (void) des_free_context(ctx);
778         plaintext->cd_length = 0;
```



```
820 /* EXPORT DELETE END */

780         return (CRYPTO_SUCCESS);
781 }

783 /* ARGSUSED */
784 static int
785 des_encrypt_atomic(crypto_provider_handle_t provider,
786     crypto_session_id_t session_id, crypto_mechanism_t *mechanism,
787     crypto_key_t *key, crypto_data_t *plaintext, crypto_data_t *ciphertext,
788     crypto_spi_ctx_template_t template, crypto_req_handle_t req)
789 {
790         int ret;

834 /* EXPORT DELETE START */

792         des_ctx_t des_ctx;              /* on the stack */
793         des_strength_t strength;
794         off_t saved_offset;
795         size_t saved_length;

797         DES_ARG_INPLACE(plaintext, ciphertext);

799         /*
800          * Plaintext must be a multiple of the block size.
801          * This test only works for non-padded mechanisms
802          * when blocksize is 2^N.
803          */
804         if ((plaintext->cd_length & (DES_BLOCK_LEN - 1)) != 0)
805                 return (CRYPTO_DATA_LEN_RANGE);

807         /* return length needed to store the output */
808         if (ciphertext->cd_length < plaintext->cd_length) {
809                 ciphertext->cd_length = plaintext->cd_length;
810                 return (CRYPTO_BUFFER_TOO_SMALL);
811         }

813         /* Check mechanism type and parameter length */
814         switch (mechanism->cm_type) {
815         case DES_ECB_MECH_INFO_TYPE:
816         case DES_CBC_MECH_INFO_TYPE:
817                 if (mechanism->cm_param_len > 0 &&
818                     mechanism->cm_param_len != DES_BLOCK_LEN)
819                         return (CRYPTO_MECHANISM_PARAM_INVALID);
820                 if (key->ck_length != DES_MINBITS)
821                         return (CRYPTO_KEY_SIZE_RANGE);
822                 strength = DES;
823                 break;
824         case DES3_ECB_MECH_INFO_TYPE:
825         case DES3_CBC_MECH_INFO_TYPE:
826                 if (mechanism->cm_param_len > 0 &&
827                     mechanism->cm_param_len != DES_BLOCK_LEN)
828                         return (CRYPTO_MECHANISM_PARAM_INVALID);
829                 if (key->ck_length != DES3_MAXBITS)
830                         return (CRYPTO_KEY_SIZE_RANGE);
831                 strength = DES3;
832                 break;
833         default:
834                 return (CRYPTO_MECHANISM_INVALID);
835         }

837         bzero(&des_ctx, sizeof (des_ctx_t));

839         if ((ret = des_common_init_ctx(&des_ctx, template, mechanism, key,
840             strength, crypto_kmflag(req))) != CRYPTO_SUCCESS) {
```

```
841                 return (ret);
842         }

844         saved_offset = ciphertext->cd_offset;
845         saved_length = ciphertext->cd_length;

847         /*
848          * Do the update on the specified input data.
849          */
850         switch (plaintext->cd_format) {
851         case CRYPTO_DATA_RAW:
852                 ret = crypto_update_iov(&des_ctx, plaintext, ciphertext,
853                     des_encrypt_contiguous_blocks, des_copy_block64);
854                 break;
855         case CRYPTO_DATA_UIO:
856                 ret = crypto_update_uio(&des_ctx, plaintext, ciphertext,
857                     des_encrypt_contiguous_blocks, des_copy_block64);
858                 break;
859         case CRYPTO_DATA_MBLK:
860                 ret = crypto_update_mp(&des_ctx, plaintext, ciphertext,
861                     des_encrypt_contiguous_blocks, des_copy_block64);
862                 break;
863         default:
864                 ret = CRYPTO_ARGUMENTS_BAD;
865         }

867         if (des_ctx.dc_flags & PROVIDER_OWNS_KEY_SCHEDULE) {
868                 bzero(des_ctx.dc_keysched, des_ctx.dc_keysched_len);
869                 kmem_free(des_ctx.dc_keysched, des_ctx.dc_keysched_len);
870         }

872         if (ret == CRYPTO_SUCCESS) {
873                 ASSERT(des_ctx.dc_remainder_len == 0);
874                 if (plaintext != ciphertext)
875                         ciphertext->cd_length =
876                             ciphertext->cd_offset - saved_offset;
877         } else {
878                 ciphertext->cd_length = saved_length;
879         }
880         ciphertext->cd_offset = saved_offset;

926 /* EXPORT DELETE END */

882         /* LINTED */
883         return (ret);
884 }

886 /* ARGSUSED */
887 static int
888 des_decrypt_atomic(crypto_provider_handle_t provider,
889     crypto_session_id_t session_id, crypto_mechanism_t *mechanism,
890     crypto_key_t *key, crypto_data_t *ciphertext, crypto_data_t *plaintext,
891     crypto_spi_ctx_template_t template, crypto_req_handle_t req)
892 {
893         int ret;

941 /* EXPORT DELETE START */

895         des_ctx_t des_ctx;      /* on the stack */
896         des_strength_t strength;
897         off_t saved_offset;
898         size_t saved_length;

900         DES_ARG_INPLACE(ciphertext, plaintext);

902         /*
```



```
903          * Ciphertext must be a multiple of the block size.
904          * This test only works for non-padded mechanisms
905          * when blocksize is 2^N.
906          */
907         if ((ciphertext->cd_length & (DES_BLOCK_LEN - 1)) != 0)
908                 return (CRYPTO_DATA_LEN_RANGE);

910         /* return length needed to store the output */
911         if (plaintext->cd_length < ciphertext->cd_length) {
912                 plaintext->cd_length = ciphertext->cd_length;
913                 return (CRYPTO_BUFFER_TOO_SMALL);
914         }

916         /* Check mechanism type and parameter length */
917         switch (mechanism->cm_type) {
918         case DES_ECB_MECH_INFO_TYPE:
919         case DES_CBC_MECH_INFO_TYPE:
920                 if (mechanism->cm_param_len > 0 &&
921                     mechanism->cm_param_len != DES_BLOCK_LEN)
922                         return (CRYPTO_MECHANISM_PARAM_INVALID);
923                 if (key->ck_length != DES_MINBITS)
924                         return (CRYPTO_KEY_SIZE_RANGE);
925                 strength = DES;
926                 break;
927         case DES3_ECB_MECH_INFO_TYPE:
928         case DES3_CBC_MECH_INFO_TYPE:
929                 if (mechanism->cm_param_len > 0 &&
930                     mechanism->cm_param_len != DES_BLOCK_LEN)
931                         return (CRYPTO_MECHANISM_PARAM_INVALID);
932                 if (key->ck_length != DES3_MAXBITS)
933                         return (CRYPTO_KEY_SIZE_RANGE);
934                 strength = DES3;
935                 break;
936         default:
937                 return (CRYPTO_MECHANISM_INVALID);
938         }

940         bzero(&des_ctx, sizeof (des_ctx_t));

942         if ((ret = des_common_init_ctx(&des_ctx, template, mechanism, key,
943             strength, crypto_kmflag(req))) != CRYPTO_SUCCESS) {
944                 return (ret);
945         }

947         saved_offset = plaintext->cd_offset;
948         saved_length = plaintext->cd_length;

950         /*
951          * Do the update on the specified input data.
952          */
953         switch (ciphertext->cd_format) {
954         case CRYPTO_DATA_RAW:
955                 ret = crypto_update_iov(&des_ctx, ciphertext, plaintext,
956                     des_decrypt_contiguous_blocks, des_copy_block64);
957                 break;
958         case CRYPTO_DATA_UIO:
959                 ret = crypto_update_uio(&des_ctx, ciphertext, plaintext,
960                     des_decrypt_contiguous_blocks, des_copy_block64);
961                 break;
962         case CRYPTO_DATA_MBLK:
963                 ret = crypto_update_mp(&des_ctx, ciphertext, plaintext,
964                     des_decrypt_contiguous_blocks, des_copy_block64);
965                 break;
966         default:
967                 ret = CRYPTO_ARGUMENTS_BAD;
968         }
```

```
 970         if (des_ctx.dc_flags & PROVIDER_OWNS_KEY_SCHEDULE) {
 971                 bzero(des_ctx.dc_keysched, des_ctx.dc_keysched_len);
 972                 kmem_free(des_ctx.dc_keysched, des_ctx.dc_keysched_len);
 973         }

 975         if (ret == CRYPTO_SUCCESS) {
 976                 ASSERT(des_ctx.dc_remainder_len == 0);
 977                 if (ciphertext != plaintext)
 978                         plaintext->cd_length =
 979                             plaintext->cd_offset - saved_offset;
 980         } else {
 981                 plaintext->cd_length = saved_length;
 982         }
 983         plaintext->cd_offset = saved_offset;

1033 /* EXPORT DELETE END */

 985         /* LINTED */
 986         return (ret);
 987 }

 989 /*
 990  * KCF software provider context template entry points.
 991  */
 992 /* ARGSUSED */
 993 static int
 994 des_create_ctx_template(crypto_provider_handle_t provider,
 995     crypto_mechanism_t *mechanism, crypto_key_t *key,
 996     crypto_spi_ctx_template_t *tmpl, size_t *tmpl_size, crypto_req_handle_t req)
 997 {

1049 /* EXPORT DELETE START */

 999         des_strength_t strength;
1000         void *keysched;
1001         size_t size;
1002         int rv;

1004         switch (mechanism->cm_type) {
1005         case DES_ECB_MECH_INFO_TYPE:
1006                 strength = DES;
1007                 break;
1008         case DES_CBC_MECH_INFO_TYPE:
1009                 strength = DES;
1010                 break;
1011         case DES3_ECB_MECH_INFO_TYPE:
1012                 strength = DES3;
1013                 break;
1014         case DES3_CBC_MECH_INFO_TYPE:
1015                 strength = DES3;
1016                 break;
1017         default:
1018                 return (CRYPTO_MECHANISM_INVALID);
1019         }

1021         if ((keysched = des_alloc_keysched(&size, strength,
1022             crypto_kmflag(req))) == NULL) {
1023                 return (CRYPTO_HOST_MEMORY);
1024         }

1026         /*
1027          * Initialize key schedule.  Key length information is stored
1028          * in the key.
1029          */
1030         if ((rv = init_keysched(key, keysched, strength)) != CRYPTO_SUCCESS) {
```



```
1031                 bzero(keysched, size);
1032                 kmem_free(keysched, size);
1033                 return (rv);
1034         }

1036         *tmpl = keysched;
1037         *tmpl_size = size;

1091 /* EXPORT DELETE END */

1039         return (CRYPTO_SUCCESS);
1040 }

1042 /* ARGSUSED */
1043 static int
1044 des_free_context(crypto_ctx_t *ctx)
1045 {

1101 /* EXPORT DELETE START */

1046         des_ctx_t *des_ctx = ctx->cc_provider_private;

1048         if (des_ctx != NULL) {
1049                 if (des_ctx->dc_flags & PROVIDER_OWNS_KEY_SCHEDULE) {
1050                         ASSERT(des_ctx->dc_keysched_len != 0);
1051                         bzero(des_ctx->dc_keysched, des_ctx->dc_keysched_len);
1052                         kmem_free(des_ctx->dc_keysched,
1053                             des_ctx->dc_keysched_len);
1054                 }
1055                 crypto_free_mode_ctx(des_ctx);
1056                 ctx->cc_provider_private = NULL;
1057         }

1116 /* EXPORT DELETE END */

1059         return (CRYPTO_SUCCESS);
1060 }

1062 /*
1063  * Pass it to des_keycheck() which will
1064  * fix it (parity bits), and check if the fixed key is weak.
1065  */
1066 /* ARGSUSED */
1067 static int
1068 des_key_check(crypto_provider_handle_t pd, crypto_mechanism_t *mech,
1069     crypto_key_t *key)
1070 {

1131 /* EXPORT DELETE START */

1071         int expectedkeylen;
1072         des_strength_t strength;
1073         uint8_t keydata[DES3_MAX_KEY_LEN];

1075         if ((mech == NULL) || (key == NULL))
1076                 return (CRYPTO_ARGUMENTS_BAD);

1078         switch (mech->cm_type) {
1079         case DES_ECB_MECH_INFO_TYPE:
1080         case DES_CBC_MECH_INFO_TYPE:
1081                 expectedkeylen = DES_MINBITS;
1082                 strength = DES;
1083                 break;
1084         case DES3_ECB_MECH_INFO_TYPE:
1085         case DES3_CBC_MECH_INFO_TYPE:
1086                 expectedkeylen = DES3_MAXBITS;
```

```
1087                 strength = DES3;
1088                 break;
1089         default:
1090                 return (CRYPTO_MECHANISM_INVALID);
1091         }

1093         if (key->ck_format != CRYPTO_KEY_RAW)
1094                 return (CRYPTO_KEY_TYPE_INCONSISTENT);

1096         if (key->ck_length != expectedkeylen)
1097                 return (CRYPTO_KEY_SIZE_RANGE);

1099         bcopy(key->ck_data, keydata, CRYPTO_BITS2BYTES(expectedkeylen));

1101         if (des_keycheck(keydata, strength, key->ck_data) == B_FALSE)
1102                 return (CRYPTO_WEAK_KEY);

1166 /* EXPORT DELETE END */

1104         return (CRYPTO_SUCCESS);
1105 }

1107 /* ARGSUSED */
1108 static int
1109 des_common_init_ctx(des_ctx_t *des_ctx, crypto_spi_ctx_template_t *template,
1110     crypto_mechanism_t *mechanism, crypto_key_t *key, des_strength_t strength,
1111     int kmflag)
1112 {
1113         int rv = CRYPTO_SUCCESS;

1179 /* EXPORT DELETE START */

1115         void *keysched;
1116         size_t size;

1118         if (template == NULL) {
1119                 if ((keysched = des_alloc_keysched(&size, strength,
1120                     kmflag)) == NULL)
1121                         return (CRYPTO_HOST_MEMORY);
1122                 /*
1123                  * Initialize key schedule.
1124                  * Key length is stored in the key.
1125                  */
1126                 if ((rv = init_keysched(key, keysched,
1127                     strength)) != CRYPTO_SUCCESS)
1128                         kmem_free(keysched, size);

1130                 des_ctx->dc_flags |= PROVIDER_OWNS_KEY_SCHEDULE;
1131                 des_ctx->dc_keysched_len = size;
1132         } else {
1133                 keysched = template;
1134         }
1135         des_ctx->dc_keysched = keysched;

1137         if (strength == DES3) {
1138                 des_ctx->dc_flags |= DES3_STRENGTH;
1139         }

1141         switch (mechanism->cm_type) {
1142         case DES_CBC_MECH_INFO_TYPE:
1143         case DES3_CBC_MECH_INFO_TYPE:
1144                 rv = cbc_init_ctx((cbc_ctx_t *)des_ctx, mechanism->cm_param,
1145                     mechanism->cm_param_len, DES_BLOCK_LEN, des_copy_block64);
1146                 break;
1147         case DES_ECB_MECH_INFO_TYPE:
1148         case DES3_ECB_MECH_INFO_TYPE:
```



```
1149                 des_ctx->dc_flags |= ECB_MODE;
1150         }

1152         if (rv != CRYPTO_SUCCESS) {
1153                 if (des_ctx->dc_flags & PROVIDER_OWNS_KEY_SCHEDULE) {
1154                         bzero(keysched, size);
1155                         kmem_free(keysched, size);
1156                 }
1157         }

1225 /* EXPORT DELETE END */

1159         return (rv);
1160 }
```

_____unchanged_portion_omitted_

```
**********************************************************
    8744 Thu Jul 11 01:29:54 2013
new/usr/src/uts/common/des/des_soft.c
first pass
**********************************************************
/*
 * 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 1989 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*      Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989 AT&T     */
/*        All Rights Reserved   */

/*
 * Portions of this source code were derived from Berkeley 4.3 BSD
 * under license from the Regents of the University of California.
 */

#ident  "%Z%%M% %I%     %E% SMI"

/*
 * Warning!  Things are arranged very carefully in this file to
 * allow read-only data to be moved to the text segment.  The
 * various DES tables must appear before any function definitions
 * (this is arranged by including them immediately below) and partab
 * must also appear before and function definitions
 * This arrangement allows all data up through the first text to
 * be moved to text.
 */

/*
 * Fast (?) software implementation of DES
 * Has been seen going at 2000 bytes/sec on a Sun-2
 * Works on a VAX too.
 * Won't work without 8 bit chars and 32 bit longs
 */

#include <sys/types.h>
#include <des/des.h>
#include <des/softdes.h>
#include <des/desdata.h>
#include <sys/debug.h>

static void des_setkey(u_char userkey[8], struct deskeydata *kd,
    unsigned int dir);
static void des_encrypt(u_char *data, struct deskeydata *kd);
```



```
/* EXPORT DELETE START */
#define btst(k, b)      (k[b >> 3] & (0x80 >> (b & 07)))
#define BIT28   (1<<28)
/* EXPORT DELETE END */

/*
 * Software encrypt or decrypt a block of data (multiple of 8 bytes)
 * Do the CBC ourselves if needed.
 */
/* ARGSUSED */
int
_des_crypt(char *buf, size_t len, struct desparams *desp)
{
/* EXPORT DELETE START */
        short i;
        uint_t mode;
        uint_t dir;
        char nextiv[8];
        struct deskeydata softkey;

        mode = desp->des_mode;
        dir = desp->des_dir;
        des_setkey(desp->des_key, &softkey, dir);
        while (len != 0) {
                switch (mode) {
                case CBC:
                        switch (dir) {
                        case ENCRYPT:
                                for (i = 0; i < 8; i++)
                                        buf[i] ^= desp->des_ivec[i];
                                des_encrypt((u_char *)buf, &softkey);
                                for (i = 0; i < 8; i++)
                                        desp->des_ivec[i] = buf[i];
                                break;
                        case DECRYPT:
                                for (i = 0; i < 8; i++)
                                        nextiv[i] = buf[i];
                                des_encrypt((u_char *)buf, &softkey);
                                for (i = 0; i < 8; i++) {
                                        buf[i] ^= desp->des_ivec[i];
                                        desp->des_ivec[i] = nextiv[i];
                                }
                                break;
                        }
                        break;
                case ECB:
                        des_encrypt((u_char *)buf, &softkey);
                        break;
                }
                buf += 8;
                len -= 8;
        }
/* EXPORT DELETE END */
        return (1);
}


/*
 * Set the key and direction for an encryption operation
 * We build the 16 key entries here
 */
/* ARGSUSED */
static void
des_setkey(u_char userkey[8], struct deskeydata *kd, unsigned int dir)
{
```

```
128 /* EXPORT DELETE START */
124         int32_t C, D;
125         short i;

127         /*
128          * First, generate C and D by permuting
129          * the key. The low order bit of each
130          * 8-bit char is not used, so C and D are only 28
131          * bits apiece.
132          */
133         {
134                 short bit;
135                 short *pcc = (short *)PC1_C, *pcd = (short *)PC1_D;

137                 C = D = 0;
138                 for (i = 0; i < 28; i++) {
139                         C <<= 1;
140                         D <<= 1;
141                         bit = *pcc++;
142                         if (btst(userkey, bit))
143                                 C |= 1;
144                         bit = *pcd++;
145                         if (btst(userkey, bit))
146                                 D |= 1;
147                 }
148         }
149         /*
150          * To generate Ki, rotate C and D according
151          * to schedule and pick up a permutation
152          * using PC2.
153          */
154         for (i = 0; i < 16; i++) {
155                 chunk_t *c;
156                 short j, k, bit;
157                 int bbit;

159                 /*
160                  * Do the "left shift" (rotate)
161                  * We know we always rotate by either 1 or 2 bits
162                  * the shifts table tells us if its 2
163                  */
164                 C <<= 1;
165                 if (C & BIT28)
166                         C |= 1;
167                 D <<= 1;
168                 if (D & BIT28)
169                         D |= 1;
170                 if (shifts[i]) {
171                         C <<= 1;
172                         if (C & BIT28)
173                                 C |= 1;
174                         D <<= 1;
175                         if (D & BIT28)
176                                 D |= 1;
177                 }
178                 /*
179                  * get Ki. Note C and D are concatenated.
180                  */
181                 bit = 0;
182                 switch (dir) {
183                 case ENCRYPT:
184                         c = &kd->keyval[i];
185                         break;
186                 case DECRYPT:
187                         c = &kd->keyval[15 - i];
188                         break;
```



```
189                 }
190                 c->long0 = 0;
191                 c->long1 = 0;
192                 bbit = (1 << 5) << 24;
193                 for (j = 0; j < 4; j++) {
194                         for (k = 0; k < 6; k++) {
195                                 if (C & (BIT28 >> PC2_C[bit]))
196                                         c->long0 |= bbit >> k;
197                                 if (D & (BIT28 >> PC2_D[bit]))
198                                         c->long1 |= bbit >> k;
199                                 bit++;
200                         }
201                         bbit >>= 8;
202                 }
203         }
209 /* EXPORT DELETE END */
204 }



208 /*
209  * Do an encryption operation
210  * Much pain is taken (with preprocessor) to avoid loops so the compiler
211  * can do address arithmetic instead of doing it at runtime.
212  * Note that the byte-to-chunk conversion is necessary to guarantee
213  * processor byte-order independence.
214  */
215 /* ARGSUSED */
216 static void
217 des_encrypt(u_char *data, struct deskeydata *kd)
218 {
225 /* EXPORT DELETE START */
219         chunk_t work1, work2;

221         /*
222          * Initial permutation
223          * and byte to chunk conversion
224          */
225         {
226                 const uint32_t *lp;
227                 uint32_t l0, l1, w;
228                 short i, pbit;

230                 work1.byte0 = data[0];
231                 work1.byte1 = data[1];
232                 work1.byte2 = data[2];
233                 work1.byte3 = data[3];
234                 work1.byte4 = data[4];
235                 work1.byte5 = data[5];
236                 work1.byte6 = data[6];
237                 work1.byte7 = data[7];
238                 l0 = l1 = 0;
239                 w = work1.long0;
240                 for (lp = &longtab[0], i = 0; i < 32; i++) {
241                         if (w & *lp++) {
242                                 pbit = IPtab[i];
243                                 if (pbit < 32)
244                                         l0 |= longtab[pbit];
245                                 else
246                                         l1 |= longtab[pbit-32];
247                         }
248                 }
249                 w = work1.long1;
250                 for (lp = &longtab[0], i = 32; i < 64; i++) {
251                         if (w & *lp++) {
252                                 pbit = IPtab[i];
```

```
253                                 if (pbit < 32)
254                                         l0 |= longtab[pbit];
255                                 else
256                                         l1 |= longtab[pbit-32];
257                         }
258                 }
259                 work2.long0 = l0;
260                 work2.long1 = l1;
261         }

263 /*
264  * Expand 8 bits of 32 bit R to 48 bit R
265  */
266 #ifdef __STDC__
267 #define do_R_to_ER(op, b) {                                     \
268         struct R_to_ER *p =                                     \
269             (struct R_to_ER *)&R_to_ER_tab[b][R.byte##b];       \
270         e0 op p->l0;                                            \
271         e1 op p->l1;                                            \
272 }
```

*unchanged_portion_omitted_*

```
327         /*
328          * Apply the 16 ciphering steps
329          */
330         {
331                 u_int r0, l0, r1, l1;

333                 l0 = work2.long0;
334                 r0 = work2.long1;
335                 cipher(0, r0, l0, r1, l1);
336                 cipher(1, r1, l1, r0, l0);
337                 cipher(2, r0, l0, r1, l1);
338                 cipher(3, r1, l1, r0, l0);
339                 cipher(4, r0, l0, r1, l1);
340                 cipher(5, r1, l1, r0, l0);
341                 cipher(6, r0, l0, r1, l1);
342                 cipher(7, r1, l1, r0, l0);
343                 cipher(8, r0, l0, r1, l1);
344                 cipher(9, r1, l1, r0, l0);
345                 cipher(10, r0, l0, r1, l1);
346                 cipher(11, r1, l1, r0, l0);
347                 cipher(12, r0, l0, r1, l1);
348                 cipher(13, r1, l1, r0, l0);
349                 cipher(14, r0, l0, r1, l1);
350                 cipher(15, r1, l1, r0, l0);
351                 work1.long0 = r0;
352                 work1.long1 = l0;
353         }

355         /*
356          * Final permutation
357          * and chunk to byte conversion
358          */
359         {
360                 const uint32_t *lp;
361                 uint32_t l0, l1, w;
362                 short i, pbit;

364                 l0 = l1 = 0;
365                 w = work1.long0;
366                 for (lp = &longtab[0], i = 0; i < 32; i++) {
367                         if (w & *lp++) {
368                                 pbit = FPtab[i];
369                                 if (pbit < 32)
370                                         l0 |= longtab[pbit];
```



```
371                                 else
372                                         l1 |= longtab[pbit-32];
373                         }
374                 }
375                 w = work1.long1;
376                 for (lp = &longtab[0], i = 32; i < 64; i++) {
377                         if (w & *lp++) {
378                                 pbit = FPtab[i];
379                                 if (pbit < 32)
380                                         l0 |= longtab[pbit];
381                                 else
382                                         l1 |= longtab[pbit-32];
383                         }
384                 }
385                 work2.long0 = l0;
386                 work2.long1 = l1;
387         }
388         data[0] = work2.byte0;
389         data[1] = work2.byte1;
390         data[2] = work2.byte2;
391         data[3] = work2.byte3;
392         data[4] = work2.byte4;
393         data[5] = work2.byte5;
394         data[6] = work2.byte6;
395         data[7] = work2.byte7;
403 /* EXPORT DELETE END */
396 }
```

*unchanged_portion_omitted_*

```
**********************************************************
   55191 Thu Jul 11 01:29:54 2013
new/usr/src/uts/common/des/desdata.h
first pass
**********************************************************
```

```
   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  * Copyright 2003 Sun Microsystems, Inc.  All rights reserved.
  23  * Use is subject to license terms.
  24  */

  26 /*      Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989 AT&T     */
  27 /*        All Rights Reserved   */

  29 /*
  30  * Portions of this source code were derived from Berkeley 4.3 BSD
  31  * under license from the Regents of the University of California.
  32  */

  34 #ifndef _SYS_DESDATA_H
  35 #define _SYS_DESDATA_H

  37 #pragma ident   "%Z%%M% %I%     %E% SMI"

  39 #ifdef  __cplusplus
  40 extern "C" {
  41 #endif

  43 /*
  44  * softdesdata.c, Data for software implementation of DES
  45  */

  47 /*
  48  * Lint can't handle static's in include files.
  49  * Complains "defined but not used" and then "used but not defined"
  50  */
  51 #ifdef __lint
  52 #define static
  53 #endif

  55 /* EXPORT DELETE START */
  55 /*
  56  * Permuted-choice 1 from the key bits
  57  * to yield C and D.
  58  * Note that bits 8,16... are left out:
  59  * They are intended for a parity check.
  60  * Table has been munged to be zero-origin
```



```
  61  */

  63 const short     PC1_C[] = {
  64         57-1, 49-1, 41-1, 33-1, 25-1, 17-1,  9-1,
  65         1-1,  58-1, 50-1, 42-1, 34-1, 26-1, 18-1,
  66         10-1,  2-1, 59-1, 51-1, 43-1, 35-1, 27-1,
  67         19-1, 11-1,  3-1, 60-1, 52-1, 44-1, 36-1,
  68 };
______unchanged_portion_omitted_
1084 /* EXPORT DELETE END */

1084 #ifdef  __cplusplus
1085 }
______unchanged_portion_omitted_
```

```
**********************************************************
    2273 Thu Jul 11 01:29:55 2013
new/usr/src/uts/common/gssapi/Makefile
first pass
**********************************************************
#
# 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 (c) 1989,1997,1999 by Sun Microsystems, Inc.
# All rights reserved.
#
# Copyright 2012 Milan Jurik. All rights reserved.
#
# uts/common/gssd/Makefile
#
# include global definitions
include ../../../Makefile.master


INSTALLED_HDRS= gssapi.h gssapi_ext.h
PRIVATE_HDRS=   gssd.x gssd_prot.h
HDRS=   $(INSTALLED_HDRS) $(PRIVATE_HDRS)

DERIVED_FILES=  gssd_prot.h gssd_prot.c gssd_xdr.c

GSSDDIRS= $(ROOT)/usr/include/gssapi

GSSDHDRS= $(INSTALLED_HDRS:%=$(GSSDDIRS)/%)

CHECKHDRS= $(INSTALLED_HDRS:%.h=%.check)

# gssd_prot.h is rpcgen'ed and can never be made to pass
# cstyle so it is unchecked
UNCHECKED_HDRS= gss_prot.h

# install rule
$(GSSDDIRS)/%: %
        $(INS.file)

.KEEP_STATE:

.PARALLEL: $(CHECKHDRS)

install_h: all_h $(GSSDDIRS) $(GSSDHDRS)

all_h: $(DERIVED_FILES)
```



```
$(GSSDDIRS):
        $(INS.dir)

gssd_prot.h:    gssd.x
        $(RM) $@
        $(RPCGEN) -CM -h gssd.x > $@

gssd_prot.c:   gssd.x
        $(RM) $@

#       Over ticotsord we do zero retries. Over ticlts we do 5
#       retries. Hence, a default of 25 seconds for ticotsord is
#       too little. 125 = 25 + 6 * MAXTIMO (from clnt_clts.c).
#
        $(RPCGEN) -M -l gssd.x | sed -e \
                's;#include..gssd.h.;#include "gssd_prot.h";' \
                | sed 's/TIMEOUT/gssd_timeout/' \
                | sed 's/{ 25, 0 }/{ 125, 0 }/' \
                | grep -v stdlib.h | grep -v stdio.h > $@

gssd_xdr.c:    gssd.x
        $(RM) $@
        $(RPCGEN) -M -c gssd.x | sed -e \
                's;#include..gssd.h.;#include "gssd_prot.h";' > $@

check:  $(CHECKHDRS)

clean:
        $(RM) $(DERIVED_FILES)

# EXPORT DELETE START
# Special target to clean up the source tree for export distribution
# Warning: This target changes the source tree
EXPORT_SRC:
        $(RM) Makefile+ gssd.x+ gssd_clnt_stubs.c+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < gssd.x > gssd.x+
        $(MV) gssd.x+ gssd.x
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < gssd_clnt_stubs.c > gssd_clnt_stubs.c+
        $(MV) gssd_clnt_stubs.c+ gssd_clnt_stubs.c
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile gssd.x gssd_clnt_stubs.c

# CRYPT DELETE START
# Special target to clean up the source tree for domestic distribution
# Warning: This target changes the source tree

CRYPT_SRC:
        $(RM) Makefile+ gssd_clnt_stubs.c+
        sed  -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        sed -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
                < gssd_clnt_stubs.c > gssd_clnt_stubs.c+
        $(MV) gssd_clnt_stubs.c+ gssd_clnt_stubs.c
        $(CHMOD) 444 Makefile gssd_clnt_stubs.c

# CRYPT DELETE END
# EXPORT DELETE END
```

```
**********************************************************
   16407 Thu Jul 11 01:29:56 2013
new/usr/src/uts/common/gssapi/gssd.x
first pass
**********************************************************
______unchanged_portion_omitted_

 268 %/* EXPORT DELETE START */
 268 struct gss_seal_arg {
 269         OM_UINT32       gssd_context_verifier;  /* verifier for context handles
 270         GSS_CTX_ID_T    context_handle;         /* handle to existing context */
 271         int             conf_req_flag;          /* type of conf requested */
 272         int             qop_req;                /* quality of prot. requested */
 273         GSS_BUFFER_T    input_message_buffer;   /* message to protect */
 274 };
______unchanged_portion_omitted_

 298 %/* EXPORT DELETE END */

 297 struct gss_display_status_arg {
 298         uid_t           uid;                    /* client uid */
 299         int             status_value;           /* status to be converted */
 300         int             status_type;            /* GSS or mech status */
 301         GSS_OID         mech_type;              /* mechanism */
 302         OM_UINT32       message_context;        /* recursion flag */
 303 };
______unchanged_portion_omitted_


 403 /*
 404  *  The server accepts requests only from the loopback address.
 405  *  Unix authentication is used, and the port must be in the reserved range.
 406  */

 408 program GSSPROG {
 409     version GSSVERS {

 411         /*
 412          *  Called by the client to acquire a credential.
 413          */
 414         gss_acquire_cred_res
 415                 GSS_ACQUIRE_CRED(gss_acquire_cred_arg)                  = 1;

 417         /*
 418          *  Called by the client to release a credential.
 419          */
 420         gss_release_cred_res
 421                 GSS_RELEASE_CRED(gss_release_cred_arg)                  = 2;

 423         /*
 424          *  Called by the client to initialize a security context.
 425          */
 426         gss_init_sec_context_res
 427                 GSS_INIT_SEC_CONTEXT(gss_init_sec_context_arg)          = 3;

 429         /*
 430          *  Called by the server to initialize a security context.
 431          */
 432         gss_accept_sec_context_res
 433                 GSS_ACCEPT_SEC_CONTEXT(gss_accept_sec_context_arg)      = 4;

 435         /*
 436          *  Called to pass token to underlying mechanism.
 437          */
 438         gss_process_context_token_res
 439                 GSS_PROCESS_CONTEXT_TOKEN(gss_process_context_token_arg) = 5;
```



```
 441         /*
 442          *  Called to delete a security context.
 443          */
 444         gss_delete_sec_context_res
 445                 GSS_DELETE_SEC_CONTEXT(gss_delete_sec_context_arg)      = 6;

 447         /*
 448          *  Called to get remaining time security context has to live.
 449          */
 450         gss_context_time_res
 451                 GSS_CONTEXT_TIME(gss_context_time_arg)                  = 7;

 453         /*
 454          *  Called to sign a message.
 455          */
 456         gss_sign_res    GSS_SIGN(gss_sign_arg)                          = 8;

 458         /*
 459          *  Called to verify a signed message.
 460          */
 461         gss_verify_res  GSS_VERIFY(gss_verify_arg)                      = 9;

 463         /*
 464          *  Called to translate minor status into a string.
 465          */
 466         gss_display_status_res
 467                         GSS_DISPLAY_STATUS(gss_display_status_arg)      = 10;

 469         /*
 470          *  Called to indicate which underlying mechanisms are supported
 471          */
 472         gss_indicate_mechs_res
 473                         GSS_INDICATE_MECHS(void)                        = 11;

 475         /*
 476          *  Called by the client to inquire about a credential.
 477          */
 478         gss_inquire_cred_res
 479                 GSS_INQUIRE_CRED(gss_inquire_cred_arg)                  = 12;


 485 /* EXPORT DELETE START */

 482         /*
 483          *  Called to seal a message.
 484          */
 485         gss_seal_res    GSS_SEAL(gss_seal_arg)                          = 13;

 487         /*
 488          *  Called to unseal a message.
 489          */
 490         gss_unseal_res  GSS_UNSEAL(gss_unseal_arg)                      = 14;

 497 /* EXPORT DELETE END */

 492         /*
 493          * gsscred interface functions to obtain principal uid and gids
 494          */
 495         gsscred_expname_to_unix_cred_res
 496                         GSSCRED_EXPNAME_TO_UNIX_CRED(
 497                                 gsscred_expname_to_unix_cred_arg)       = 15;

 499         gsscred_name_to_unix_cred_res
 500                         GSSCRED_NAME_TO_UNIX_CRED(
 501                                 gsscred_name_to_unix_cred_arg)          = 16;
```

```
503         gss_get_group_info_res
504                         GSS_GET_GROUP_INFO(gss_get_group_info_arg)      = 17;

506         gss_get_kmod_res
507                         GSS_GET_KMOD(gss_get_kmod_arg)                  = 18;

509         gss_export_sec_context_res
510                         GSS_EXPORT_SEC_CONTEXT(gss_export_sec_context_arg) = 19;
511
512         gss_import_sec_context_res
513                         GSS_IMPORT_SEC_CONTEXT(gss_import_sec_context_arg) = 20;
514         /*
515          *  Called by the client to add to a credential.
516          */
517         gss_add_cred_res
518                 GSS_ADD_CRED(gss_add_cred_arg)                          = 21;
519         gss_inquire_cred_by_mech_res
520                 GSS_INQUIRE_CRED_BY_MECH(gss_inquire_cred_by_mech_arg)
521                                                                         = 22;

523     } = 1;
524 } = 100234;
```

unchanged_portion_omitted_

```
**********************************************************
   74047 Thu Jul 11 01:29:56 2013
new/usr/src/uts/common/gssapi/gssd_clnt_stubs.c
first pass
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 * Copyright 2012 Milan Jurik. All rights reserved.
 */

/*
 *  GSSAPI library stub module for gssd.
 */

#include <mechglueP.h>
#include "gssd_prot.h"
#include <rpc/rpc.h>

#include <sys/systm.h>
#include <sys/types.h>
#include <sys/cmn_err.h>
#include <sys/kmem.h>
#include <gssapi/kgssapi_defs.h>
#include <sys/debug.h>

#ifdef GSSDEBUG
/*
 * Kernel kgssd module debugging aid. The global variable "gss_log"
 * is a bit mask which allows various types of debugging messages
 * to be printed out.
 *
 *       gss_log & 1  will cause actual failures to be printed.
 *       gss_log & 2  will cause informational messages to be
 *                       printed on the client side of kgssd.
 *       gss_log & 4  will cause informational messages to be
 *                       printed on the server side of kgssd.
 *       gss_log & 8  will cause informational messages to be
 *                       printed on both client and server side of kgssd.
 */

uint_t gss_log = 1;

#endif /* GSSDEBUG */

#ifdef  DEBUG
```



```
extern void prom_printf(const char *, ...);
#endif

char *server = "localhost";

static OM_uint32 kgss_sign_wrapped(void *, OM_uint32 *, gss_ctx_id_t, int,
        gss_buffer_t, gss_buffer_t,  OM_uint32);

static OM_uint32 kgss_verify_wrapped(void *, OM_uint32 *, gss_ctx_id_t,
        gss_buffer_t, gss_buffer_t, int *qop_state, OM_uint32);

/* EXPORT DELETE START */
static OM_uint32 kgss_seal_wrapped(void *, OM_uint32 *, gss_ctx_id_t,
        int, int, gss_buffer_t, int *, gss_buffer_t,  OM_uint32);

static OM_uint32 kgss_unseal_wrapped(void *, OM_uint32 *, gss_ctx_id_t,
        gss_buffer_t, gss_buffer_t, int *conf_state, int *qop_state,
        OM_uint32);
/* EXPORT DELETE END */

static OM_uint32 kgss_delete_sec_context_wrapped(void *, OM_uint32 *,
        gssd_ctx_id_t *, gss_buffer_t, OM_uint32);

static void __kgss_reset_mech(gss_mechanism *, gss_OID);

#define DEFAULT_MINOR_STAT      ((OM_uint32) ~0)

OM_uint32
kgss_acquire_cred_wrapped(minor_status,
                        desired_name,
                        time_req,
                        desired_mechs,
                        cred_usage,
                        output_cred_handle,
                        actual_mechs,
                        time_rec,
                        uid,
                        gssd_cred_verifier)
        OM_uint32 *minor_status;
        const gss_name_t desired_name;
        OM_uint32 time_req;
        const gss_OID_set desired_mechs;
        int cred_usage;
        gssd_cred_id_t *output_cred_handle;
        gss_OID_set *actual_mechs;
        OM_uint32 *time_rec;
        uid_t uid;
        OM_uint32 *gssd_cred_verifier;
{
        CLIENT *clnt;

        OM_uint32       minor_status_temp;
        gss_buffer_desc external_name;
        gss_OID         name_type;
        enum clnt_stat  client_stat;
        int             i;

        gss_acquire_cred_arg arg;
        gss_acquire_cred_res res;

        /* get the client handle to GSSD */

        if ((clnt = getgssd_handle()) == NULL) {
                GSSLOG(1, "kgss_acquire_cred: can't connect to server on %s\n",
                        server);
                return (GSS_S_FAILURE);
```

```
126         }
128         /* convert the desired name from internal to external format */
130         if (gss_display_name(&minor_status_temp, desired_name, &external_name,
131                              &name_type) != GSS_S_COMPLETE) {
133                 *minor_status = (OM_uint32) minor_status_temp;
134                 killgssd_handle(clnt);
135                 GSSLOG0(1, "kgss_acquire_cred: display name failed\n");
136                 return ((OM_uint32) GSS_S_FAILURE);
137         }
140         /* copy the procedure arguments into the rpc arg parameter */
142         arg.uid = (OM_uint32) uid;
144         arg.desired_name.GSS_BUFFER_T_len = (uint_t)external_name.length;
145         arg.desired_name.GSS_BUFFER_T_val = (char *)external_name.value;
147         arg.name_type.GSS_OID_len =
148                 name_type == GSS_C_NULL_OID ?
149                         0 : (uint_t)name_type->length;
151         arg.name_type.GSS_OID_val =
152                 name_type == GSS_C_NULL_OID ?
153                         (char *)NULL : (char *)name_type->elements;
155         arg.time_req = time_req;
157         if (desired_mechs != GSS_C_NULL_OID_SET) {
158                 arg.desired_mechs.GSS_OID_SET_len =
159                         (uint_t)desired_mechs->count;
160                 arg.desired_mechs.GSS_OID_SET_val = (GSS_OID *)
161                         MALLOC(sizeof (GSS_OID) * desired_mechs->count);
163                 for (i = 0; i < desired_mechs->count; i++) {
164                     arg.desired_mechs.GSS_OID_SET_val[i].GSS_OID_len =
165                         (uint_t)desired_mechs->elements[i].length;
166                     arg.desired_mechs.GSS_OID_SET_val[i].GSS_OID_val =
167                         (char *)MALLOC(desired_mechs->elements[i].length);
168                     (void) memcpy(
169                         arg.desired_mechs.GSS_OID_SET_val[i].GSS_OID_val,
170                         desired_mechs->elements[i].elements,
171                         desired_mechs->elements[i].length);
172                 }
173         } else
174                 arg.desired_mechs.GSS_OID_SET_len = 0;
176         arg.cred_usage = cred_usage;
178         /* call the remote procedure */
180         bzero((caddr_t)&res, sizeof (res));
181         client_stat = gss_acquire_cred_1(&arg, &res, clnt);
183         (void) gss_release_buffer(&minor_status_temp, &external_name);
184         if (desired_mechs != GSS_C_NULL_OID_SET) {
185                 for (i = 0; i < desired_mechs->count; i++)
186                         FREE(arg.desired_mechs.GSS_OID_SET_val[i].GSS_OID_val,
187                             arg.desired_mechs.GSS_OID_SET_val[i].GSS_OID_len);
188                 FREE(arg.desired_mechs.GSS_OID_SET_val,
189                     arg.desired_mechs.GSS_OID_SET_len * sizeof (GSS_OID));
190         }
```



```
192         if (client_stat != RPC_SUCCESS) {
194                 /*
195                  * if the RPC call times out, null out all return arguments,
196                  * set minor_status to its maximum value, and return
197                  * GSS_S_FAILURE
198                  */
200                 if (minor_status != NULL)
201                         *minor_status = DEFAULT_MINOR_STAT;
202                 if (output_cred_handle != NULL)
203                         *output_cred_handle = NULL;
204                 if (actual_mechs != NULL)
205                         *actual_mechs = NULL;
206                 if (time_rec != NULL)
207                         *time_rec = 0;
209                 killgssd_handle(clnt);
210                 GSSLOG0(1, "kgss_acquire_cred: RPC call times out\n");
211                 return (GSS_S_FAILURE);
212         }
214         /* copy the rpc results into the return arguments */
216         if (minor_status != NULL)
217                 *minor_status = res.minor_status;
219         if (output_cred_handle != NULL &&
220                 (res.status == GSS_S_COMPLETE)) {
221             *output_cred_handle =
222                 *((gssd_cred_id_t *)res.output_cred_handle.GSS_CRED_ID_T_val);
223             *gssd_cred_verifier = res.gssd_cred_verifier;
224         }
226         if (res.status == GSS_S_COMPLETE &&
227                 res.actual_mechs.GSS_OID_SET_len != 0 &&
228                 actual_mechs != NULL) {
229                 *actual_mechs = (gss_OID_set) MALLOC(sizeof (gss_OID_set_desc));
230                 (*actual_mechs)->count =
231                                         (int)res.actual_mechs.GSS_OID_SET_len;
232                 (*actual_mechs)->elements = (gss_OID)
233                         MALLOC(sizeof (gss_OID_desc) * (*actual_mechs)->count);
235                 for (i = 0; i < (*actual_mechs)->count; i++) {
236                     (*actual_mechs)->elements[i].length = (OM_uint32)
237                         res.actual_mechs.GSS_OID_SET_val[i].GSS_OID_len;
238                     (*actual_mechs)->elements[i].elements =
239                         (void *) MALLOC((*actual_mechs)->elements[i].length);
240                     (void) memcpy((*actual_mechs)->elements[i].elements,
241                         res.actual_mechs.GSS_OID_SET_val[i].GSS_OID_val,
242                         (*actual_mechs)->elements[i].length);
243                 }
244         } else {
245                 if (res.status == GSS_S_COMPLETE &&
246                         actual_mechs != NULL)
247                         (*actual_mechs) = NULL;
248         }
250         if (time_rec != NULL)
251                 *time_rec = res.time_rec;
253         /*
254          * free the memory allocated for the results and return with the status
255          * received in the rpc call
256          */
```

```
 258         clnt_freeres(clnt, xdr_gss_acquire_cred_res, (caddr_t)&res);
 259         killgssd_handle(clnt);
 260         return (res.status);

 262 }
______unchanged_portion_omitted_

 817 static struct gss_config default_gc = {
 818         { 0, NULL},
 819         NULL,
 820         NULL,
 821         0,
 824 /* EXPORT DELETE START */ /* CRYPT DELETE START */
 822         kgss_unseal_wrapped,
 826 /* EXPORT DELETE END */ /* CRYPT DELETE END */
 823         NULL,           /* kgss_delete_sec_context_wrapped */
 828 /* EXPORT DELETE START */ /* CRYPT DELETE START */
 824         kgss_seal_wrapped,
 830 /* EXPORT DELETE END */ /* CRYPT DELETE END */
 825         NULL,           /* kgss_import_sec_context */
 832 /* EXPORT DELETE START */
 833 /* CRYPT DELETE START */
 834 #if 0
 835 /* CRYPT DELETE END */
 836         kgss_seal_wrapped,
 837         kgss_unseal_wrapped,
 838 /* CRYPT DELETE START */
 839 #endif
 840 /* CRYPT DELETE END */
 841 /* EXPORT DELETE END */
 826         kgss_sign_wrapped,
 827         kgss_verify_wrapped
 828 };
______unchanged_portion_omitted_

1792 /* EXPORT DELETE START */

1776 /*ARGSUSED*/
1777 static OM_uint32
1778 kgss_seal_wrapped(void *private,
1779         OM_uint32 *minor_status,
1780         const gss_ctx_id_t ctx_handle,
1781         int conf_req_flag,
1782         int qop_req,
1783         const gss_buffer_t input_message_buffer,
1784         int *conf_state,
1785         gss_buffer_t output_message_buffer,
1786         OM_uint32 gssd_context_verifier)
1787 {
1788         CLIENT *clnt;
1789         gssd_ctx_id_t   context_handle;

1791         gss_seal_arg arg;
1792         gss_seal_res res;

1794         context_handle = (gssd_ctx_id_t)KCTX_TO_GSSD_CTX(ctx_handle);

1796         /* get the client handle to GSSD */

1798         if ((clnt = getgssd_handle()) == NULL) {
1799                 GSSLOG(1, "kgss_seal: can't connect to server on %s\n", server);
1800                 return (GSS_S_FAILURE);
1801         }

1803         /* copy the procedure arguments into the rpc arg parameter */
```



```
1805         arg.context_handle.GSS_CTX_ID_T_len = (uint_t)sizeof (gss_ctx_id_t);
1806         arg.context_handle.GSS_CTX_ID_T_val = (char *)&context_handle;

1808         arg.context_handle.GSS_CTX_ID_T_len = (uint_t)sizeof (OM_uint32);
1809         arg.context_handle.GSS_CTX_ID_T_val = (char *)&context_handle;
1810         arg.gssd_context_verifier = gssd_context_verifier;

1812         arg.conf_req_flag = conf_req_flag;

1814         arg.qop_req = qop_req;

1816         arg.input_message_buffer.GSS_BUFFER_T_len =
1817             (uint_t)input_message_buffer->length;

1819         arg.input_message_buffer.GSS_BUFFER_T_val =
1820             (char *)input_message_buffer->value;

1822         /* call the remote procedure */

1824         bzero((caddr_t)&res, sizeof (res));
1825         if (gss_seal_1(&arg, &res, clnt) != RPC_SUCCESS) {

1827         /*
1828          * if the RPC call times out, null out all return arguments, set
1829          * minor_status to its maximum value, and return GSS_S_FAILURE
1830          */

1832                 if (minor_status != NULL)
1833                         *minor_status = DEFAULT_MINOR_STAT;
1834                 if (conf_state != NULL)
1835                         *conf_state = 0;
1836                 if (output_message_buffer != NULL)
1837                         output_message_buffer->length = 0;

1839                 killgssd_handle(clnt);
1840                 GSSLOG0(1, "kgss_seal: RPC call times out\n");
1841                 return (GSS_S_FAILURE);
1842         }

1844         /* copy the rpc results into the return arguments */

1846         if (minor_status != NULL)
1847                 *minor_status = res.minor_status;

1849         if (conf_state != NULL)
1850                 *conf_state = res.conf_state;

1852         if (output_message_buffer != NULL) {
1853                 output_message_buffer->length =
1854                     res.output_message_buffer.GSS_BUFFER_T_len;

1856                 output_message_buffer->value =
1857                     (void *) MALLOC(output_message_buffer->length);
1858                 (void) memcpy(output_message_buffer->value,
1859                     res.output_message_buffer.GSS_BUFFER_T_val,
1860                     output_message_buffer->length);
1861         }

1863         /*
1864          * free the memory allocated for the results and return with the status
1865          * received in the rpc call
1866          */

1868         clnt_freeres(clnt, xdr_gss_seal_res, (caddr_t)&res);
1869         killgssd_handle(clnt);
1870         return (res.status);
```

```
1871 }
_____unchanged_portion_omitted_

2024 /* EXPORT DELETE END */

2006 OM_uint32
2007 kgss_display_status(minor_status,
2008                 status_value,
2009                 status_type,
2010                 mech_type,
2011                 message_context,
2012                 status_string,
2013                 uid)
2014         OM_uint32 *minor_status;
2015         OM_uint32 status_value;
2016         int status_type;
2017         const gss_OID mech_type;
2018         int *message_context;
2019         gss_buffer_t status_string;
2020         uid_t uid;
2021 {
2022         CLIENT *clnt;

2024         gss_display_status_arg arg;
2025         gss_display_status_res res;

2027         /* get the client handle to GSSD */

2029         if ((clnt = getgssd_handle()) == NULL) {
2030         GSSLOG(1, "kgss_display_status: can't connect to server on %s\n",
2031                         server);
2032                 return (GSS_S_FAILURE);
2033         }

2035         /* copy the procedure arguments into the rpc arg parameter */

2037         arg.uid = (OM_uint32) uid;

2039         arg.status_value = status_value;
2040         arg.status_type = status_type;

2042         arg.mech_type.GSS_OID_len = (uint_t)(mech_type != GSS_C_NULL_OID ?
2043                                              mech_type->length : 0);
2044         arg.mech_type.GSS_OID_val = (char *)(mech_type != GSS_C_NULL_OID ?
2045                                              mech_type->elements : 0);

2047         arg.message_context = *message_context;

2049         /* call the remote procedure */

2051         if (message_context != NULL)
2052                 *message_context = 0;
2053         if (status_string != NULL) {
2054                 status_string->length = 0;
2055                 status_string->value = NULL;
2056         }

2058         bzero((caddr_t)&res, sizeof (res));
2059         if (gss_display_status_1(&arg, &res, clnt) != RPC_SUCCESS) {

2061         /*
2062          * if the RPC call times out, null out all return arguments, set
2063          * minor_status to its maximum value, and return GSS_S_FAILURE
2064          */

2066                 if (minor_status != NULL)
```



```
2067                         *minor_status = DEFAULT_MINOR_STAT;

2069                 killgssd_handle(clnt);
2070                 GSSLOG0(1, "kgss_display_status: RPC call time out\n");
2071                 return (GSS_S_FAILURE);
2072         }


2075         /* now process the results and pass them back to the caller */

2077         if (res.status == GSS_S_COMPLETE) {
2078                 if (minor_status != NULL)
2079                         *minor_status = res.minor_status;
2080                 if (message_context != NULL)
2081                         *message_context = res.message_context;
2082                 if (status_string != NULL) {
2083                         status_string->length =
2084                                 (size_t)res.status_string.GSS_BUFFER_T_len;
2085                         status_string->value =
2086                                 (void *) MALLOC(status_string->length);
2087                         (void) memcpy(status_string->value,
2088                                 res.status_string.GSS_BUFFER_T_val,
2089                                 status_string->length);
2090                 }
2091         }

2093         clnt_freeres(clnt, xdr_gss_display_status_res, (caddr_t)&res);
2094         killgssd_handle(clnt);
2095         return (res.status);
2096 }
_____unchanged_portion_omitted_
```

```
**********************************************************
    1006 Thu Jul 11 01:29:57 2013
new/usr/src/uts/common/gssapi/include/Makefile
first pass
**********************************************************
#
# 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 (c) 1997-2001 by Sun Microsystems, Inc.
# All rights reserved.
#
#pragma ident   "%Z%%M% %I%     %E% SMI"

# include global definitions
include ../../../../Makefile.master

# EXPORT DELETE START

all:
        @$(ECHO) " This Makefile is used to clean up the source tree\n" \
                "for export distribution.\n" \
                "[Usage]: make [EXPORT_SRC] [CRYPT_SRC]\n\n" \
                "WARNING: EXPORT_SRC, CRYPT_SRC targets change the\n" \
                "source tree and remove the Makefile."

# Special target to clean up the source tree for export distribution
# Warning: This target changes the source tree and removes this Makefile

EXPORT_SRC:
        $(RM) gssapiP_dummy.h+ mechglueP.h+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < gssapiP_dummy.h > gssapiP_dummy.h+
        $(MV) gssapiP_dummy.h+ gssapiP_dummy.h
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < mechglueP.h > mechglueP.h+
        $(MV) mechglueP.h+ mechglueP.h
        $(RM) Makefile
        $(CHMOD) 444 gssapiP_dummy.h mechglueP.h

# CRYPT DELETE START
# Special target to clean up the source tree for export distribution
# Warning: This target changes the source tree.

CRYPT_SRC:
        $(RM) mechglueP.h+ Makefile+
        sed -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
                < mechglueP.h > mechglueP.h+
```



```
        $(MV) mechglueP.h+ mechglueP.h
        sed -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 mechglueP.h Makefile

# CRYPT DELETE END
# EXPORT DELETE END
```

```
**********************************************************
   11379 Thu Jul 11 01:29:57 2013
new/usr/src/uts/common/gssapi/include/gssapiP_dummy.h
first pass
**********************************************************
______unchanged_portion_omitted_
```

```
  64 const gss_OID_set_desc * const gss_mech_set_dummy = dummy_oidsets+0;

  66 #define TWRITE_STR(ptr, str, len) \
  67         (void) memcpy((ptr), (char *) (str), (len)); \
  68         (ptr) += (len);
  69 #ifndef _KERNEL

  71 #ifdef DEBUG_ON

  73 #define dprintf(a) printf(a)
  74 #define dprintf1(a, b) printf(a, b)

  76 #else

  78 #define dprintf(a)
  79 #define dprintf1(a, b)
  80 #define DUMMY_STATIC

  82 #endif  /* DEBUG_ON */

  84 #else   /* _KERNEL */

  86 #if defined(DEBUG) && !defined(DUMMY_MECH_DEBUG)
  87 #define DUMMY_MECH_DEBUG
  88 #endif

  90 #ifdef DUMMY_MECH_DEBUG
  91 #define DUMMY_MECH_LOG(A, B, C) \
  92         ((void)((dummy_mech_log & (A)) && (printf((B), (C)), TRUE)))
  93 #define DUMMY_MECH_LOG0(A, B)   \
  94         ((void)((dummy_mech_log & (A)) && (printf(B), TRUE)))
  95 #else
  96 #define DUMMY_MECH_LOG(A, B, C)
  97 #define DUMMY_MECH_LOG0(A, B)

  99 #endif

 101 #define dprintf(a)      DUMMY_MECH_LOG0(8, a)
 102 #define dprintf1(a, b)  DUMMY_MECH_LOG(8, a, b)
 103 #define DUMMY_STATIC    static

 105 #endif  /* _KERNEL */

 107 /*
 108  * declarations of internal name mechanism functions
 109  */

 111 OM_uint32 dummy_gss_acquire_cred
 112         (
 113                 void *,                 /* dummy context */
 114                 OM_uint32 *,            /* minor_status */
 115                 gss_name_t,             /* desired_name */
 116                 OM_uint32,              /* time_req */
 117                 gss_OID_set,            /* desired_mechs */
 118                 gss_cred_usage_t,       /* cred_usage */
 119                 gss_cred_id_t *,        /* output_cred_handle */
 120                 gss_OID_set *,          /* actual_mechs */
 121                 OM_uint32 *             /* time_rec */
 122         /* */);
```



```
 124 OM_uint32 dummy_gss_release_cred
 125         (
 126                 void *,                 /* dummy context */
 127                 OM_uint32 *,            /* minor_status */
 128                 gss_cred_id_t *         /* cred_handle */
 129         /* */);

 131 OM_uint32 dummy_gss_init_sec_context
 132         (
 133                 void *,                 /* dummy context */
 134                 OM_uint32 *,            /* minor_status */
 135                 gss_cred_id_t,          /* claimant_cred_handle */
 136                 gss_ctx_id_t *,         /* context_handle */
 137                 gss_name_t,             /* target_name */
 138                 gss_OID,                /* mech_type */
 139                 OM_uint32,              /* req_flags */
 140                 OM_uint32,              /* time_req */
 141                 gss_channel_bindings_t, /* input_chan_bindings */
 142                 gss_buffer_t,           /* input_token */
 143                 gss_OID *,              /* actual_mech_type */
 144                 gss_buffer_t,           /* output_token */
 145                 OM_uint32 *,            /* ret_flags */
 146                 OM_uint32 *             /* time_rec */
 147         /* */);

 149 OM_uint32 dummy_gss_accept_sec_context
 150         (
 151                 void *,                 /* dummy context */
 152                 OM_uint32 *,            /* minor_status */
 153                 gss_ctx_id_t *,         /* context_handle */
 154                 gss_cred_id_t,          /* verifier_cred_handle */
 155                 gss_buffer_t,           /* input_token_buffer */
 156                 gss_channel_bindings_t, /* input_chan_bindings */
 157                 gss_name_t *,           /* src_name */
 158                 gss_OID *,              /* mech_type */
 159                 gss_buffer_t,           /* output_token */
 160                 OM_uint32 *,            /* ret_flags */
 161                 OM_uint32 *,            /* time_rec */
 162                 gss_cred_id_t *         /* delegated_cred_handle */
 163         /* */);

 165 OM_uint32 dummy_gss_process_context_token
 166         (
 167                 void *,                 /* dummy context */
 168                 OM_uint32 *,            /* minor_status */
 169                 gss_ctx_id_t,           /* context_handle */
 170                 gss_buffer_t            /* token_buffer */
 171         /* */);

 173 DUMMY_STATIC OM_uint32 dummy_gss_delete_sec_context
 174         (
 175                 void *,                 /* dummy context */
 176                 OM_uint32 *,            /* minor_status */
 177                 gss_ctx_id_t *,         /* context_handle */
 178                 gss_buffer_t            /* output_token */
 179 #ifdef  _KERNEL
 180         /* */, OM_uint32
 181 #endif
 182         /* */);

 184 OM_uint32 dummy_gss_context_time
 185         (
 186                 void *,                 /* dummy context */
 187                 OM_uint32 *,            /* minor_status */
 188                 gss_ctx_id_t,           /* context_handle */
 189                 OM_uint32 *             /* time_rec */
```

```
190         /* */);

192 DUMMY_STATIC OM_uint32 dummy_gss_sign
193         (
194                 void *,                 /* dummy context */
195                 OM_uint32 *,            /* minor_status */
196                 gss_ctx_id_t,           /* context_handle */
197                 int,                    /* qop_req */
198                 gss_buffer_t,           /* message_buffer */
199                 gss_buffer_t            /* message_token */
200 #ifdef  _KERNEL
201         /* */, OM_uint32
202 #endif
203         /* */);

205 DUMMY_STATIC OM_uint32 dummy_gss_verify
206         (
207                 void *,                 /* dummy context */
208                 OM_uint32 *,            /* minor_status */
209                 gss_ctx_id_t,           /* context_handle */
210                 gss_buffer_t,           /* message_buffer */
211                 gss_buffer_t,           /* token_buffer */
212                 int *                   /* qop_state */
213 #ifdef  _KERNEL
214         /* */, OM_uint32

216 #endif
217         /* */);

218 /* EXPORT DELETE START */
219 DUMMY_STATIC OM_uint32 dummy_gss_seal
220         (
221                 void *,                 /* dummy context */
222                 OM_uint32 *,            /* minor_status */
223                 gss_ctx_id_t,           /* context_handle */
224                 int,                    /* conf_req_flag */
225                 int,                    /* qop_req */
226                 gss_buffer_t,           /* input_message_buffer */
227                 int *,                  /* conf_state */
228                 gss_buffer_t            /* output_message_buffer */
229 #ifdef  _KERNEL
230         /* */, OM_uint32
231 #endif
232         /* */);

234 DUMMY_STATIC OM_uint32 dummy_gss_unseal
235         (
236                 void *,                 /* dummy context */
237                 OM_uint32 *,            /* minor_status */
238                 gss_ctx_id_t,           /* context_handle */
239                 gss_buffer_t,           /* input_message_buffer */
240                 gss_buffer_t,           /* output_message_buffer */
241                 int *,                  /* conf_state */
242                 int *                   /* qop_state */
243 #ifdef  _KERNEL
244         /* */, OM_uint32
245 #endif
246         /* */);
247 /* EXPORT DELETE END */

248 OM_uint32 dummy_gss_display_status
249         (
250                 void *,                 /* dummy context */
251                 OM_uint32 *,            /* minor_status */
252                 OM_uint32,              /* status_value */
253                 int,                    /* status_type */
```



```
254                 gss_OID,                /* mech_type */
255                 OM_uint32 *,            /* message_context */
256                 gss_buffer_t            /* status_string */
257         /* */);

259 OM_uint32 dummy_gss_indicate_mechs
260         (
261                 void *,                 /* dummy context */
262                 OM_uint32 *,            /* minor_status */
263                 gss_OID_set *           /* mech_set */
264         /* */);

266 OM_uint32 dummy_gss_compare_name
267         (
268                 void *,                 /* dummy context */
269                 OM_uint32 *,            /* minor_status */
270                 gss_name_t,             /* name1 */
271                 gss_name_t,             /* name2 */
272                 int *                   /* name_equal */
273         /* */);

275 OM_uint32 dummy_gss_display_name
276         (
277                 void *,                 /* dummy context */
278                 OM_uint32 *,            /* minor_status */
279                 gss_name_t,             /* input_name */
280                 gss_buffer_t,           /* output_name_buffer */
281                 gss_OID *               /* output_name_type */
282         /* */);

284 OM_uint32 dummy_gss_import_name
285         (
286                 void *,                 /* dummy context */
287                 OM_uint32 *,            /* minor_status */
288                 gss_buffer_t,           /* input_name_buffer */
289                 gss_OID,                /* input_name_type */
290                 gss_name_t *            /* output_name */
291         /* */);

293 OM_uint32 dummy_gss_release_name
294         (
295                 void *,                 /* dummy context */
296                 OM_uint32 *,            /* minor_status */
297                 gss_name_t *            /* input_name */
298         /* */);

300 OM_uint32 dummy_gss_inquire_cred
301         (
302                 void *,                 /* dummy context */
303                 OM_uint32 *,            /* minor_status */
304                 gss_cred_id_t,          /* cred_handle */
305                 gss_name_t *,           /* name */
306                 OM_uint32 *,            /* lifetime */
307                 gss_cred_usage_t *,     /* cred_usage */
308                 gss_OID_set *           /* mechanisms */
309         /* */);

311 OM_uint32 dummy_gss_inquire_context
312         (
313                 void *,                 /* dummy context */
314                 OM_uint32 *,            /* minor_status */
315                 gss_ctx_id_t,           /* context_handle */
316                 gss_name_t *,           /* initiator_name */
317                 gss_name_t *,           /* acceptor_name */
318                 OM_uint32 *,            /* lifetime_rec */
319                 gss_OID *,              /* mech_type */
```

```
320                 OM_uint32 *,            /* ret_flags */
321                 int *,                  /* locally_initiated */
322                 int *                   /* open */
323         /* */);

325 /* New V2 entry points */
326 OM_uint32 dummy_gss_get_mic
327         (
328                 void *,                 /* dummy context */
329                 OM_uint32 *,            /* minor_status */
330                 gss_ctx_id_t,           /* context_handle */
331                 gss_qop_t,              /* qop_req */
332                 gss_buffer_t,           /* message_buffer */
333                 gss_buffer_t            /* message_token */
334         /* */);

336 OM_uint32 dummy_gss_verify_mic
337         (
338                 void *,                 /* dummy context */
339                 OM_uint32 *,            /* minor_status */
340                 gss_ctx_id_t,           /* context_handle */
341                 gss_buffer_t,           /* message_buffer */
342                 gss_buffer_t,           /* message_token */
343                 gss_qop_t *             /* qop_state */
344         /* */);

346 OM_uint32 dummy_gss_wrap
347         (
348                 void *,                 /* dummy context */
349                 OM_uint32 *,            /* minor_status */
350                 gss_ctx_id_t,           /* context_handle */
351                 int,                    /* conf_req_flag */
352                 gss_qop_t,              /* qop_req */
353                 gss_buffer_t,           /* input_message_buffer */
354                 int *,                  /* conf_state */
355                 gss_buffer_t            /* output_message_buffer */
356         /* */);

358 OM_uint32 dummy_gss_unwrap
359         (
360                 void *,                 /* dummy context */
361                 OM_uint32 *,            /* minor_status */
362                 gss_ctx_id_t,           /* context_handle */
363                 gss_buffer_t,           /* input_message_buffer */
364                 gss_buffer_t,           /* output_message_buffer */
365                 int *,                  /* conf_state */
366                 gss_qop_t *             /* qop_state */
367         /* */);

369 OM_uint32 dummy_gss_wrap_size_limit
370         (
371                 void *,                 /* dummy context */
372                 OM_uint32 *,            /* minor_status */
373                 gss_ctx_id_t,           /* context_handle */
374                 int,                    /* conf_req_flag */
375                 gss_qop_t,              /* qop_req */
376                 OM_uint32,              /* req_output_size */
377                 OM_uint32 *             /* max_input_size */
378         /* */);

380 OM_uint32 dummy_gss_add_cred
381         (
382                 void *,                 /* dummy context */
383                 OM_uint32 *,            /* minor_status */
384                 gss_cred_id_t,          /* input_cred_handle */
385                 gss_name_t,             /* desired_name */
```



```
386                 gss_OID,                /* desired_mech */
387                 gss_cred_usage_t,       /* cred_usage */
388                 OM_uint32,              /* initiator_time_req */
389                 OM_uint32,              /* acceptor_time_req */
390                 gss_cred_id_t *,        /* output_cred_handle */
391                 gss_OID_set *,          /* actual_mechs */
392                 OM_uint32 *,            /* initiator_time_rec */
393                 OM_uint32 *             /* acceptor_time_rec */
394         /* */);

396 OM_uint32 dummy_gss_inquire_cred_by_mech
397         (
398                 void *,                 /* dummy context */
399                 OM_uint32  *,           /* minor_status */
400                 gss_cred_id_t,          /* cred_handle */
401                 gss_OID,                /* mech_type */
402                 gss_name_t *,           /* name */
403                 OM_uint32 *,            /* initiator_lifetime */
404                 OM_uint32 *,            /* acceptor_lifetime */
405                 gss_cred_usage_t *      /* cred_usage */
406         /* */);

408 OM_uint32 dummy_gss_export_sec_context
409         (
410                 void *,                 /* dummy context */
411                 OM_uint32 *,            /* minor_status */
412                 gss_ctx_id_t *,         /* context_handle */
413                 gss_buffer_t            /* interprocess_token */
414         /* */);

416 OM_uint32 dummy_gss_import_sec_context
417         (
418                 void *,                 /* dummy context */
419                 OM_uint32 *,            /* minor_status */
420                 gss_buffer_t,           /* interprocess_token */
421                 gss_ctx_id_t *          /* context_handle */
422         /* */);

424 #if 0
425 OM_uint32 dummy_gss_release_oid
426         (
427                 OM_uint32 *,            /* minor_status */
428                 gss_OID *               /* oid */
429         /* */);
430 #endif

432 OM_uint32 dummy_gss_internal_release_oid
433         (
434                 void *,                 /* dummy context  */
435                 OM_uint32 *,            /* minor_status */
436                 gss_OID *               /* oid */
437         /* */);

439 OM_uint32 dummy_gss_inquire_names_for_mech
440         (
441                 void *,         /* dummy context */
442                 OM_uint32 *,            /* minor_status */
443                 gss_OID,                /* mechanism */
444                 gss_OID_set *           /* name_types */
445         /* */);

447 OM_uint32 dummy_pname_to_uid
448         (
449                 void *,                 /* dummy context */
450                 OM_uint32 *,            /* minor status */
451                 const gss_name_t,       /* pname */
```

```
 452                 uid_t *                 /* uidOut */
 453         /* */);


 456 #ifdef  __cplusplus
 457 }
```

*unchanged_portion_omitted_*

```
**********************************************************
   27355 Thu Jul 11 01:29:58 2013
new/usr/src/uts/common/gssapi/include/mechglueP.h
first pass
**********************************************************
______unchanged_portion_omitted_

 116 /* Solaris Kerberos */
 117 typedef OM_uint32          (*gss_acquire_cred_with_password_sfct)(
 118                     void *,             /* context */
 119                     OM_uint32 *,        /* minor_status */
 120                     const gss_name_t,   /* desired_name */
 121                     const gss_buffer_t, /* password */
 122                     OM_uint32,          /* time_req */
 123                     const gss_OID_set,  /* desired_mechs */
 124                     int,                /* cred_usage */
 125                     gss_cred_id_t *,    /* output_cred_handle */
 126                     gss_OID_set *,      /* actual_mechs */
 127                     OM_uint32 *         /* time_rec */
 128         /* */);

 130 /*
 131  * Rudimentary pointer validation macro to check whether the
 132  * "loopback" field of an opaque struct points back to itself.  This
 133  * field also catches some programming errors where an opaque pointer
 134  * is passed to a function expecting the address of the opaque
 135  * pointer.
 136  */
 137 #if 0 /* Solaris Kerberos - revisit for full 1.7/next resync */
 138 #define GSSINT_CHK_LOOP(p) (!((p) != NULL && (p)->loopback == (p)))
 139 #else
 140 #define GSSINT_CHK_LOOP(p) ((p) == NULL)
 141 #endif


 144 /*********************************************************/
 145 /* The Mechanism Dispatch Table -- a mechanism needs to */
 146 /* define one of these and provide a function to return */
 147 /* it to initialize the GSSAPI library                  */

 149 /*
 150  * This is the definition of the mechs_array struct, which is used to
 151  * define the mechs array table. This table is used to indirectly
 152  * access mechanism specific versions of the gssapi routines through
 153  * the routines in the glue module (gssd_mech_glue.c)
 154  *
 155  * This contants all of the functions defined in gssapi.h except for
 156  * gss_release_buffer() and gss_release_oid_set(), which I am
 157  * assuming, for now, to be equal across mechanisms.
 158  */
 159
 160 typedef struct gss_config {
 161 #if 0 /* Solaris Kerberos */
 162     OM_uint32       priority;
 163     char *          mechNameStr;
 164 #endif
 165     gss_OID_desc    mech_type;
 166     void *          context;
 167 #ifdef  _KERNEL
 168     struct gss_config *next;
 169     bool_t          uses_kmod;
 170 #endif

 172 #ifndef _KERNEL
 173         OM_uint32           (*gss_acquire_cred)
 174         (
```



```
 175                     void *,             /* context */

 177                     OM_uint32 *,        /* minor_status */
 178                     const gss_name_t,   /* desired_name */
 179                     OM_uint32,          /* time_req */
 180                     const gss_OID_set,  /* desired_mechs */
 181                     int,                /* cred_usage */
 182                     gss_cred_id_t *,    /* output_cred_handle */
 183                     gss_OID_set *,      /* actual_mechs */
 184                     OM_uint32 *         /* time_rec */
 185         /* */);
 186         OM_uint32           (*gss_release_cred)
 187         (

 189                     void *,             /* context */
 190                     OM_uint32 *,        /* minor_status */
 191                     gss_cred_id_t *     /* cred_handle */
 192         /* */);
 193         OM_uint32           (*gss_init_sec_context)
 194         (
 195                     void *,                     /* context */
 196                     OM_uint32 *,                /* minor_status */
 197                     const gss_cred_id_t,        /* claimant_cred_handle */
 198                     gss_ctx_id_t *,             /* context_handle */
 199                     const gss_name_t,           /* target_name */
 200                     const gss_OID,              /* mech_type */
 201                     OM_uint32,                  /* req_flags */
 202                     OM_uint32,                  /* time_req */
 203                     const gss_channel_bindings_t, /* input_chan_bindings */
 204                     const gss_buffer_t,         /* input_token */
 205                     gss_OID*,                   /* actual_mech_type */
 206                     gss_buffer_t,               /* output_token */
 207                     OM_uint32 *,                /* ret_flags */
 208                     OM_uint32 *                 /* time_rec */
 209         /* */);
 210         OM_uint32           (*gss_accept_sec_context)
 211         (
 212                     void *,                     /* context */
 213                     OM_uint32 *,                /* minor_status */
 214                     gss_ctx_id_t *,             /* context_handle */
 215                     const gss_cred_id_t,        /* verifier_cred_handle */
 216                     const gss_buffer_t,         /* input_token_buffer */
 217                     const gss_channel_bindings_t, /* input_chan_bindings */
 218                     gss_name_t *,               /* src_name */
 219                     gss_OID*,                   /* mech_type */
 220                     gss_buffer_t,               /* output_token */
 221                     OM_uint32 *,                       /* ret_flags */
 222                     OM_uint32 *,                       /* time_rec */
 223                     gss_cred_id_t *             /* delegated_cred_handle */
 224         /* */);
 225 /* EXPORT DELETE START */ /* CRYPT DELETE START */
 225 #endif  /* ! _KERNEL */

 227 /*
 228  * Note: there are two gss_unseal's in here. Make any changes to both.
 229  */
 230         OM_uint32           (*gss_unseal)
 231         (
 232                     void *,             /* context */
 233                     OM_uint32 *,        /* minor_status */
 234                     const gss_ctx_id_t, /* context_handle */
 235                     const gss_buffer_t, /* input_message_buffer */
 236                     gss_buffer_t,       /* output_message_buffer */
 237                     int *,              /* conf_state */
 238                     int *               /* qop_state */
 239 #ifdef   _KERNEL
```

```
240         /* */, OM_uint32
241 #endif
242         /* */);
243 #ifndef _KERNEL
245 /* EXPORT DELETE END */ /* CRYPT DELETE END */
244         OM_uint32           (*gss_process_context_token)
245         (
246                     void *,             /* context */
247                     OM_uint32 *,        /* minor_status */
248                     const gss_ctx_id_t, /* context_handle */
249                     const gss_buffer_t  /* token_buffer */
250         /* */);
251 #endif  /* ! _KERNEL */
252         OM_uint32           (*gss_delete_sec_context)
253         (
254                     void *,             /* context */
255                     OM_uint32 *,        /* minor_status */
256                     gss_ctx_id_t *,     /* context_handle */
257                     gss_buffer_t        /* output_token */
258 #ifdef   _KERNEL
259         /* */, OM_uint32
260 #endif
261         /* */);
262 #ifndef _KERNEL
263         OM_uint32           (*gss_context_time)
264         (
265                     void *,             /* context */
266                     OM_uint32 *,        /* minor_status */
267                     const gss_ctx_id_t, /* context_handle */
268                     OM_uint32 *         /* time_rec */
269         /* */);
270         OM_uint32           (*gss_display_status)
271         (
272                     void *,             /* context */
273                     OM_uint32 *,        /* minor_status */
274                     OM_uint32,          /* status_value */
275                     int,                /* status_type */
276                     const gss_OID,      /* mech_type */
277                     OM_uint32 *,        /* message_context */
278                     gss_buffer_t        /* status_string */
279         /* */);
280         OM_uint32           (*gss_indicate_mechs)
281         (
282                     void *,             /* context */
283                     OM_uint32 *,        /* minor_status */
284                     gss_OID_set *       /* mech_set */
285         /* */);
286         OM_uint32           (*gss_compare_name)
287         (
288                     void *,             /* context */
289                     OM_uint32 *,        /* minor_status */
290                     const gss_name_t,   /* name1 */
291                     const gss_name_t,   /* name2 */
292                     int *               /* name_equal */
293         /* */);
294         OM_uint32           (*gss_display_name)
295         (
296                     void *,             /* context */
297                     OM_uint32 *,        /* minor_status */
298                     const gss_name_t,   /* input_name */
299                     gss_buffer_t,       /* output_name_buffer */
300                     gss_OID*            /* output_name_type */
301         /* */);
302         OM_uint32           (*gss_import_name)
303         (
304                     void *,             /* context */
```



```
305                     OM_uint32 *,        /* minor_status */
306                     const gss_buffer_t, /* input_name_buffer */
307                     const gss_OID,      /* input_name_type */
308                     gss_name_t *        /* output_name */
309         /* */);
310         OM_uint32           (*gss_release_name)
311         (
312                     void *,             /* context */
313                     OM_uint32 *,        /* minor_status */
314                     gss_name_t *        /* input_name */
315         /* */);
316         OM_uint32           (*gss_inquire_cred)
317         (
318                     void *,                     /* context */
319                     OM_uint32 *,                /* minor_status */
320                     const gss_cred_id_t,        /* cred_handle */
321                     gss_name_t *,               /* name */
322                     OM_uint32 *,                /* lifetime */
323                     int *,                      /* cred_usage */
324                     gss_OID_set *               /* mechanisms */
325         /* */);
326         OM_uint32           (*gss_add_cred)
327         (
328                     void *,             /* context */
329                     OM_uint32 *,        /* minor_status */
330                     const gss_cred_id_t,        /* input_cred_handle */
331                     const gss_name_t,   /* desired_name */
332                     const gss_OID,      /* desired_mech */
333                     gss_cred_usage_t,   /* cred_usage */
334                     OM_uint32,          /* initiator_time_req */
335                     OM_uint32,          /* acceptor_time_req */
336                     gss_cred_id_t *,    /* output_cred_handle */
337                     gss_OID_set *,      /* actual_mechs */
338                     OM_uint32 *,        /* initiator_time_rec */
339                     OM_uint32 *         /* acceptor_time_rec */
340         /* */);
343 /* EXPORT DELETE START */ /* CRYPT DELETE START */
341 #endif  /* ! _KERNEL */
342 /*
343  * Note: there are two gss_seal's in here. Make any changes to both.
344  */
345         OM_uint32           (*gss_seal)
346         (
347                     void *,             /* context */
348                     OM_uint32 *,        /* minor_status */
349                     const gss_ctx_id_t, /* context_handle */
350                     int,                /* conf_req_flag */
351                     int,                /* qop_req */
352                     const gss_buffer_t, /* input_message_buffer */
353                     int *,              /* conf_state */
354                     gss_buffer_t        /* output_message_buffer */
355 #ifdef   _KERNEL
356         /* */, OM_uint32
357 #endif
358         /* */);
359 #ifndef _KERNEL
363 /* EXPORT DELETE END */ /* CRYPT DELETE END */
360         OM_uint32           (*gss_export_sec_context)
361         (
362                     void *,             /* context */
363                     OM_uint32 *,        /* minor_status */
364                     gss_ctx_id_t *,     /* context_handle */
365                     gss_buffer_t        /* interprocess_token */
366         /* */);
367 #endif  /* ! _KERNEL */
368         OM_uint32           (*gss_import_sec_context)
```

```
369         (
370                     void *,             /* context */
371                     OM_uint32 *,        /* minor_status */
372                     const gss_buffer_t, /* interprocess_token */
373                     gss_ctx_id_t *      /* context_handle */
374         /* */);
375 #ifndef _KERNEL
376         OM_uint32           (*gss_inquire_cred_by_mech)
377         (
378                     void *,             /* context */
379                     OM_uint32 *,        /* minor_status */
380                     const gss_cred_id_t,        /* cred_handle */
381                     const gss_OID,      /* mech_type */
382                     gss_name_t *,       /* name */
383                     OM_uint32 *,        /* initiator_lifetime */
384                     OM_uint32 *,        /* acceptor_lifetime */
385                     gss_cred_usage_t *  /* cred_usage */
386         /* */);
387         OM_uint32           (*gss_inquire_names_for_mech)
388         (
389                     void *,             /* context */
390                     OM_uint32 *,        /* minor_status */
391                     const gss_OID,      /* mechanism */
392                     gss_OID_set *       /* name_types */
393         /* */);
394         OM_uint32       (*gss_inquire_context)
395         (
396                     void *,             /* context */
397                     OM_uint32 *,        /* minor_status */
398                     const gss_ctx_id_t, /* context_handle */
399                     gss_name_t *,       /* src_name */
400                     gss_name_t *,       /* targ_name */
401                     OM_uint32 *,        /* lifetime_rec */
402                     gss_OID *,          /* mech_type */
403                     OM_uint32 *,        /* ctx_flags */
404                     int *,              /* locally_initiated */
405                     int *               /* open */
406         /* */);
407         OM_uint32           (*gss_internal_release_oid)
408         (
409                     void *,             /* context */
410                     OM_uint32 *,        /* minor_status */
411                     gss_OID *           /* OID */
412         /* */);
413         OM_uint32               (*gss_wrap_size_limit)
414         (
415                     void *,             /* context */
416                     OM_uint32 *,        /* minor_status */
417                     const gss_ctx_id_t, /* context_handle */
418                     int,                /* conf_req_flag */
419                     gss_qop_t,          /* qop_req */
420                     OM_uint32,          /* req_output_size */
421                     OM_uint32 *         /* max_input_size */
422         /* */);
423         OM_uint32               (*pname_to_uid)
424         (
425                     void *,             /* context */
426                     OM_uint32 *,        /* minor_status */
427                     const gss_name_t,   /* pname */
428                     uid_t *             /* uid */
429         /* */);
430         OM_uint32               (*__gss_userok)
431         (
432                     void *,             /* context */
433                     OM_uint32 *,        /* minor_status */
434                     const gss_name_t,   /* pname */
```



```
435                     const char *,       /* local user */
436                     int *               /* user ok? */
437         /* */);
438         OM_uint32               (*gss_export_name)
439         (
440                 void *,                 /* context */
441                 OM_uint32 *,            /* minor_status */
442                 const gss_name_t,       /* input_name */
443                 gss_buffer_t            /* exported_name */
444         /* */);
445 #endif  /* ! _KERNEL */
446 /* EXPORT DELETE START */
447 /* CRYPT DELETE START */
448 /*
449  * This block comment is Sun Proprietary: Need-To-Know.
450  * What we are doing is leaving the seal and unseal entry points
451  * in an obvious place before sign and unsign for the Domestic customer
452  * of the Solaris Source Product. The Domestic customer of the Solaris Source
453  * Product will have to deal with the problem of creating exportable libgss
454  * binaries.
455  * In the binary product that Sun builds, these entry points are elsewhere,
456  * and bracketed with special comments so that the CRYPT_SRC and EXPORT_SRC
457  * targets delete them.
458  */
459 #if 0
460 /* CRYPT DELETE END */
461         OM_uint32           (*gss_seal)
462         (
463                     void *,             /* context */
464                     OM_uint32 *,        /* minor_status */
465                     const gss_ctx_id_t, /* context_handle */
466                     int,                /* conf_req_flag */
467                     int,                /* qop_req */
468                     const gss_buffer_t, /* input_message_buffer */
469                     int *,              /* conf_state */
470                     gss_buffer_t        /* output_message_buffer */
471 #ifdef   _KERNEL
472         /* */, OM_uint32
473 #endif
474         /* */);
475         OM_uint32           (*gss_unseal)
476         (
477                     void *,             /* context */
478                     OM_uint32 *,        /* minor_status */
479                     const gss_ctx_id_t, /* context_handle */
480                     const gss_buffer_t, /* input_message_buffer */
481                     gss_buffer_t,       /* output_message_buffer */
482                     int *,              /* conf_state */
483                     int *               /* qop_state */
484 #ifdef   _KERNEL
485         /* */, OM_uint32
486 #endif
487         /* */);
488 /* CRYPT DELETE START */
489 #endif /* 0 */
490 /* CRYPT DELETE END */
491 /* EXPORT DELETE END */
492         OM_uint32       (*gss_sign)
493         (
494                     void *,             /* context */
495                     OM_uint32 *,        /* minor_status */
496                     const gss_ctx_id_t, /* context_handle */
497                     int,                /* qop_req */
498                     const gss_buffer_t, /* message_buffer */
499                     gss_buffer_t        /* message_token */
500 #ifdef   _KERNEL
```

```
455         /* */, OM_uint32
456 #endif
457         /* */);
458         OM_uint32       (*gss_verify)
459         (
460                 void *,                 /* context */
461                 OM_uint32 *,            /* minor_status */
462                 const gss_ctx_id_t,     /* context_handle */
463                 const gss_buffer_t,     /* message_buffer */
464                 const gss_buffer_t,     /* token_buffer */
465                 int *                   /* qop_state */
466 #ifdef   _KERNEL
467         /* */, OM_uint32
468 #endif
469         /* */);
470 #ifndef  _KERNEL
471         OM_uint32       (*gss_store_cred)
472         (
473                 void *,                 /* context */
474                 OM_uint32 *,            /* minor_status */
475                 const gss_cred_id_t,    /* input_cred */
476                 gss_cred_usage_t,       /* cred_usage */
477                 const gss_OID,          /* desired_mech */
478                 OM_uint32,              /* overwrite_cred */
479                 OM_uint32,              /* default_cred */
480                 gss_OID_set *,          /* elements_stored */
481                 gss_cred_usage_t *      /* cred_usage_stored */
482         /* */);

484         /* GGF extensions */

486         OM_uint32       (*gss_inquire_sec_context_by_oid)
487         (
488                 OM_uint32 *,        /* minor_status */
489                 const gss_ctx_id_t, /* context_handle */
490                 const gss_OID,      /* OID */
491                 gss_buffer_set_t *  /* data_set */
492         /* */);

494 #endif
495 } *gss_mechanism;
```

unchanged_portion_omitted_

```
714 #define KCTX_TO_KGSS_CTX(ctx)   ((struct kgss_ctx *)(ctx))
715 #define KCTX_TO_CTX_IMPORTED(ctx)       (KCTX_TO_KGSS_CTX(ctx)->ctx_imported)
716 #define KCTX_TO_GSSD_CTX(ctx)   (KCTX_TO_KGSS_CTX(ctx)->gssd_ctx)
717 #define KCTX_TO_CTXV(ctx)       (KCTX_TO_KGSS_CTX(ctx)->gssd_ctx_verifier)
718 #define KCTX_TO_MECH(ctx)       (KCTX_TO_KGSS_CTX(ctx)->mech)
719 #define KCTX_TO_PRIVATE(ctx)    (KCTX_TO_MECH(ctx)->context)
720 #define KGSS_CTX_TO_GSSD_CTX(ctx)       \
721         (((ctx) == GSS_C_NO_CONTEXT) ? (gssd_ctx_id_t)(uintptr_t)(ctx) : \
722         KCTX_TO_GSSD_CTX(ctx))
723 #define KGSS_CTX_TO_GSSD_CTXV(ctx)      \
724         (((ctx) == GSS_C_NO_CONTEXT) ? (NULL) : KCTX_TO_CTXV(ctx))

726 #ifdef _KERNEL
727 #define KCTX_TO_I_CTX(ctx)      (KCTX_TO_KGSS_CTX(ctx)->gssd_i_ctx)
728 #define KCTX_TO_CTX(ctx) \
729 ((KCTX_TO_CTX_IMPORTED(ctx) == FALSE) ? (ctx) : \
730         KCTX_TO_I_CTX(ctx))
731 #define KGSS_CRED_ALLOC()       kmem_zalloc(sizeof (struct kgss_cred), \
732         KM_SLEEP)
733 #define KGSS_CRED_FREE(cred)    kmem_free(cred, sizeof (struct kgss_cred))

735 #define KGSS_ALLOC()    kmem_zalloc(sizeof (struct kgss_ctx), KM_SLEEP)
736 #define KGSS_FREE(ctx)  kmem_free(ctx, sizeof (struct kgss_ctx))
```



```
738 #define KGSS_SIGN(minor_st, ctx, qop, msg, tkn) \
739         (*(KCTX_TO_MECH(ctx)->gss_sign))(KCTX_TO_PRIVATE(ctx), minor_st, \
740                 KCTX_TO_CTX(ctx), qop, msg, tkn, KCTX_TO_CTXV(ctx))

742 #define KGSS_VERIFY(minor_st, ctx, msg, tkn, qop)       \
743         (*(KCTX_TO_MECH(ctx)->gss_verify))(KCTX_TO_PRIVATE(ctx), minor_st,\
744                 KCTX_TO_CTX(ctx), msg, tkn, qop,  KCTX_TO_CTXV(ctx))

746 #define KGSS_DELETE_SEC_CONTEXT(minor_st, ctx, int_ctx_id,  tkn)        \
747         (*(KCTX_TO_MECH(ctx)->gss_delete_sec_context))(KCTX_TO_PRIVATE(ctx),\
748                 minor_st, int_ctx_id, tkn, KCTX_TO_CTXV(ctx))

750 #define KGSS_IMPORT_SEC_CONTEXT(minor_st, tkn, ctx, int_ctx_id) \
751         (*(KCTX_TO_MECH(ctx)->gss_import_sec_context))(KCTX_TO_PRIVATE(ctx),\
752                 minor_st, tkn, int_ctx_id)

804 /* EXPORT DELETE START */
754 #define KGSS_SEAL(minor_st, ctx, conf_req, qop, msg, conf_state, tkn) \
755         (*(KCTX_TO_MECH(ctx)->gss_seal))(KCTX_TO_PRIVATE(ctx), minor_st, \
756                 KCTX_TO_CTX(ctx), conf_req, qop, msg, conf_state, tkn,\
757                 KCTX_TO_CTXV(ctx))

759 #define KGSS_UNSEAL(minor_st, ctx, msg, tkn, conf, qop) \
760         (*(KCTX_TO_MECH(ctx)->gss_unseal))(KCTX_TO_PRIVATE(ctx), minor_st,\
761                 KCTX_TO_CTX(ctx), msg, tkn, conf, qop, \
762                 KCTX_TO_CTXV(ctx))

815 /* EXPORT DELETE END */

764 #define KGSS_INIT_CONTEXT(ctx) krb5_init_context(ctx)
765 #define KGSS_RELEASE_OID(minor_st, oid) krb5_gss_release_oid(minor_st, oid)
766 extern OM_uint32 kgss_release_oid(OM_uint32 *, gss_OID *);

768 #else /* !_KERNEL */

770 #define KGSS_INIT_CONTEXT(ctx) krb5_gss_init_context(ctx)
771 #define KGSS_RELEASE_OID(minor_st, oid) gss_release_oid(minor_st, oid)

773 #define KCTX_TO_CTX(ctx)  (KCTX_TO_KGSS_CTX(ctx)->gssd_ctx)
774 #define MALLOC(n) malloc(n)
775 #define FREE(x, n) free(x)
776 #define KGSS_CRED_ALLOC()       (struct kgss_cred *) \
777                 MALLOC(sizeof (struct kgss_cred))
778 #define KGSS_CRED_FREE(cred)    free(cred)
779 #define KGSS_ALLOC()    (struct kgss_ctx *)MALLOC(sizeof (struct kgss_ctx))
780 #define KGSS_FREE(ctx)  free(ctx)

782 #define KGSS_SIGN(minor_st, ctx, qop, msg, tkn) \
783         kgss_sign_wrapped(minor_st, \
784                 KCTX_TO_CTX(ctx), qop, msg, tkn, KCTX_TO_CTXV(ctx))

786 #define KGSS_VERIFY(minor_st, ctx, msg, tkn, qop)       \
787         kgss_verify_wrapped(minor_st,\
788                 KCTX_TO_CTX(ctx), msg, tkn, qop, KCTX_TO_CTXV(ctx))

790 #define KGSS_SEAL(minor_st, ctx, conf_req, qop, msg, conf_state, tkn) \
791         kgss_seal_wrapped(minor_st, \
792                 KCTX_TO_CTX(ctx), conf_req, qop, msg, conf_state, tkn, \
793                 KCTX_TO_CTXV(ctx))

795 #define KGSS_UNSEAL(minor_st, ctx, msg, tkn, conf, qop) \
796         kgss_unseal_wrapped(minor_st,\
797                 KCTX_TO_CTX(ctx), msg, tkn, conf, qop,  \
798                 KCTX_TO_CTXV(ctx))
799 #endif /* _KERNEL */
```

```
801 /* SUNW15resync - moved from gssapiP_generic.h for sake of non-krb5 mechs */
802 OM_uint32 generic_gss_release_buffer
803 (OM_uint32*,       /* minor_status */
804             gss_buffer_t      /* buffer */
805           );

807 OM_uint32 generic_gss_release_oid_set
808 (OM_uint32*,       /* minor_status */
809             gss_OID_set*      /* set */
810           );

812 OM_uint32 generic_gss_release_oid
813 (OM_uint32*,       /* minor_status */
814             gss_OID*         /* set */
815           );

817 OM_uint32 generic_gss_copy_oid
818 (OM_uint32 *,   /* minor_status */
819             gss_OID_desc * const,   /* oid */ /* SUNW15resync */
820             gss_OID *           /* new_oid */
821             );

823 OM_uint32 generic_gss_create_empty_oid_set
824 (OM_uint32 *,   /* minor_status */
825             gss_OID_set *       /* oid_set */
826           );

828 OM_uint32 generic_gss_add_oid_set_member
829 (OM_uint32 *,   /* minor_status */
830             gss_OID_desc * const,              /* member_oid */
831             gss_OID_set *       /* oid_set */
832           );

834 OM_uint32 generic_gss_test_oid_set_member
835 (OM_uint32 *,   /* minor_status */
836             gss_OID_desc * const,              /* member */
837             gss_OID_set,        /* set */
838             int *               /* present */
839           );

841 OM_uint32 generic_gss_oid_to_str
842 (OM_uint32 *,   /* minor_status */
843             gss_OID_desc * const,              /* oid */
844             gss_buffer_t        /* oid_str */
845           );

847 OM_uint32 generic_gss_str_to_oid
848 (OM_uint32 *,   /* minor_status */
849             gss_buffer_t,       /* oid_str */
850             gss_OID *           /* oid */
851           );

853 OM_uint32
854 generic_gss_oid_compose(
855     OM_uint32 *,        /* minor_status */
856     const char *,       /* prefix */
857     size_t,             /* prefix_len */
858     int,                /* suffix */
859     gss_OID_desc *);    /* oid */

861 OM_uint32
862 generic_gss_oid_decompose(
863     OM_uint32 *,        /* minor_status */
864     const char *,       /*prefix */
865     size_t,             /* prefix_len */
```



```
866     gss_OID_desc *,     /* oid */
867     int *);             /* suffix */

869 OM_uint32 generic_gss_create_empty_buffer_set
870 (OM_uint32 * /*minor_status*/,
871             gss_buffer_set_t * /*buffer_set*/);

873 OM_uint32 generic_gss_add_buffer_set_member
874 (OM_uint32 * /*minor_status*/,
875             const gss_buffer_t /*member_buffer*/,
876             gss_buffer_set_t * /*buffer_set*/);

878 OM_uint32 generic_gss_release_buffer_set
879 (OM_uint32 * /*minor_status*/,
880             gss_buffer_set_t * /*buffer_set*/);

882 /*
883  * SUNW17PACresync
884  * New map error API in MIT 1.7, at build time generates code for errors.
885  * Solaris does not gen the errors at build time so we just stub these
886  * for now, need to revisit.
887  * See mglueP.h and util_errmap.c in MIT 1.7.
888 */
889 #ifdef _KERNEL

891 #define map_error(MINORP, MECH)
892 #define map_errcode(MINORP)

894 #else  /* _KERNEL */

896 /* Use this to map an error code that was returned from a mech
897    operation; the mech will be asked to produce the associated error
898    messages.

900    Remember that if the minor status code cannot be returned to the
901    caller (e.g., if it's stuffed in an automatic variable and then
902    ignored), then we don't care about producing a mapping.  */
903 #define map_error(MINORP, MECH) \
904     (*(MINORP) = gssint_mecherrmap_map(*(MINORP), &(MECH)->mech_type))
905 #define map_error_oid(MINORP, MECHOID) \
906     (*(MINORP) = gssint_mecherrmap_map(*(MINORP), (MECHOID)))

908 /* Use this to map an errno value or com_err error code being
909    generated within the mechglue code (e.g., by calling generic oid
910    ops).  Any errno or com_err values produced by mech operations
911    should be processed with map_error.  This means they'll be stored
912    separately even if the mech uses com_err, because we can't assume
913    that it will use com_err.  */
914 #define map_errcode(MINORP) \
915     (*(MINORP) = gssint_mecherrmap_map_errcode(*(MINORP)))

917 #endif /* _KERNEL */

919 #endif /* _GSS_MECHGLUEP_H */
```

```
**********************************************************
     980 Thu Jul 11 01:29:59 2013
new/usr/src/uts/common/gssapi/mechs/dummy/Makefile
first pass
**********************************************************
```

```
   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 (c) 1997-2001 by Sun Microsystems, Inc.
  24 # All rights reserved.
  25 #
  26 #pragma ident   "%Z%%M% %I%     %E% SMI"

  28 include ../../../../../Makefile.master

  30 # EXPORT DELETE START

  30 all:
  33         @$(ECHO) " This Makefile is used to clean up the source tree\n" \
  34                 "for export distribution.\n" \
  35                 "[Usage]: make [EXPORT_SRC] [CRYPT_SRC]\n\n" \
  36                 "WARNING: EXPORT_SRC, CRYPT_SRC targets change the\n" \
  37                 "source tree and remove the Makefile."

  39 # Special target to clean up the source tree for export distribution
  40 # Warning: This target changes the source tree and removes this Makefile

  42 EXPORT_SRC:
  43         $(RM) dmech.c+
  44         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
  45                 < dmech.c > dmech.c+
  46         $(MV) dmech.c+ dmech.c
  47         $(RM) Makefile
  48         $(CHMOD) 444 dmech.c

  50 # CRYPT DELETE START
  51 # CRYPT DELETE START
  52 # Special target to clean up the source tree for domestic distribution
  53 # Warning: This target changes the source tree

  55 CRYPT_SRC:
  56         $(RM) dmech.c+ Makefile+
  57         sed -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
  58                 < dmech.c > dmech.c+
  59         $(MV) dmech.c+ dmech.c
  60         sed -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d" \
  61                 < Makefile > Makefile+
```



```
  62         $(MV) Makefile+ Makefile
  63         $(CHMOD) 444 dmech.c Makefile

  65 # CRYPT DELETE END
  66 # EXPORT DELETE END
```

```
**********************************************************
   14099 Thu Jul 11 01:29:59 2013
new/usr/src/uts/common/gssapi/mechs/dummy/dmech.c
first pass
**********************************************************
```

```
/*
 * 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 2004 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 * Copyright (c) 2011 Bayard G. Bell. All rights reserved.
 */

/*
 * A module that implements a dummy security mechanism.
 * It's mainly used to test GSS-API application. Multiple tokens
 * exchanged during security context establishment can be
 * specified through dummy_mech.conf located in /etc.
 */

#include <sys/types.h>
#include <sys/modctl.h>
#include <sys/errno.h>
#include <gssapiP_dummy.h>
#include <gssapi_err_generic.h>
#include <mechglueP.h>
#include <gssapi/kgssapi_defs.h>
#include <sys/debug.h>

#ifdef DUMMY_MECH_DEBUG
/*
 * Kernel kgssd module debugging aid. The global variable "dummy_mech_log"
 * is a bit mask which allows various types of debugging messages
 * to be printed out.
 *
 *       dummy_mech_log & 1  will cause actual failures to be printed.
 *       dummy_mech_log & 2  will cause informational messages to be
 *                           printed on the client side of kgssd.
 *       dummy_mech_log & 4  will cause informational messages to be
 *                           printed on the server side of kgssd.
 *       dummy_mech_log & 8  will cause informational messages to be
 *                           printed on both client and server side of kgssd.
 */

uint_t dummy_mech_log = 1;
#endif
```



```
/* Local defines */
#define MAGIC_TOKEN_NUMBER 12345
/* private routines for dummy_mechanism */
static gss_buffer_desc make_dummy_token_msg(void *data, int datalen);

static int der_length_size(int);

static void der_write_length(unsigned char **, int);
static int der_read_length(unsigned char **, int *);
static int g_token_size(gss_OID mech, unsigned int body_size);
static void g_make_token_header(gss_OID mech, int body_size,
                                unsigned char **buf, int tok_type);
static int g_verify_token_header(gss_OID mech, int *body_size,
                                unsigned char **buf_in, int tok_type,
                                int toksize);

/* private global variables */
static int dummy_token_nums;

/*
 * This OID:
 * { iso(1) org(3) internet(6) dod(1) private(4) enterprises(1) sun(42)
 * products(2) gssapi(26) mechtypes(1) dummy(2) }
 */

static struct gss_config dummy_mechanism =
        {{10, "\053\006\001\004\001\052\002\032\001\002"},
        NULL,   /* context */
        NULL,   /* next */
        TRUE,   /* uses_kmod */
/* EXPORT DELETE START */ /* CRYPT DELETE START */
        dummy_gss_unseal,
/* EXPORT DELETE END */ /* CRYPT DELETE END */
        dummy_gss_delete_sec_context,
/* EXPORT DELETE START */ /* CRYPT DELETE START */
        dummy_gss_seal,
/* EXPORT DELETE END */ /* CRYPT DELETE END */
        dummy_gss_import_sec_context,
/* EXPORT DELETE START */
/* CRYPT DELETE START */
#if 0
/* CRYPT DELETE END */
        dummy_gss_seal,
        dummy_gss_unseal,
/* CRYPT DELETE START */
#endif
/* CRYPT DELETE END */
/* EXPORT DELETE END */
        dummy_gss_sign,
        dummy_gss_verify
};
______unchanged_portion_omitted_

/* EXPORT DELETE START */
/*ARGSUSED*/
static OM_uint32
dummy_gss_seal(context, minor_status, context_handle, conf_req_flag,
                qop_req, input_message_buffer, conf_state,
                output_message_buffer, gssd_ctx_verifier)
        void *context;
        OM_uint32 *minor_status;
        gss_ctx_id_t context_handle;
        int conf_req_flag;
        int qop_req;
        gss_buffer_t input_message_buffer;
        int *conf_state;
```

```
276         gss_buffer_t output_message_buffer;
277         OM_uint32 gssd_ctx_verifier;
278 {
279         gss_buffer_desc output;
280         dummy_gss_ctx_id_rec    *ctx;
281         dprintf("Entering gss_seal\n");

283         if (context_handle == GSS_C_NO_CONTEXT)
284                 return (GSS_S_NO_CONTEXT);
285         ctx = (dummy_gss_ctx_id_rec *) context_handle;
286         ASSERT(ctx->established == 1);
287         ASSERT(ctx->token_number == MAGIC_TOKEN_NUMBER);
288         /* Copy the input message to output message */
289         output = make_dummy_token_msg(
290                 input_message_buffer->value, input_message_buffer->length);

292         if (conf_state)
293                 *conf_state = 1;

295         *output_message_buffer = output;

297         dprintf("Leaving gss_seal\n");
298         return (GSS_S_COMPLETE);
299 }
```

_____***unchanged_portion_omitted_***

```
366 /* EXPORT DELETE END */
```

```
351 /*ARGSUSED*/
352 OM_uint32
353         dummy_gss_import_sec_context(ct, minor_status, interprocess_token,
354                                      context_handle)
355 void *ct;
356 OM_uint32 *minor_status;
357 gss_buffer_t interprocess_token;
358 gss_ctx_id_t *context_handle;
359 {
360         unsigned char *ptr;
361         int bodysize;
362         int err;

364         /* Assume that we got ctx from the interprocess token. */
365         dummy_gss_ctx_id_t ctx;

367         dprintf("Entering import_sec_context\n");
368         ptr = (unsigned char *) interprocess_token->value;
369         if (err = g_verify_token_header((gss_OID)gss_mech_dummy, &bodysize,
370                                      &ptr, 0,
371                                      interprocess_token->length)) {
372                 *minor_status = err;
373                 return (GSS_S_DEFECTIVE_TOKEN);
374         }
375         ctx = (dummy_gss_ctx_id_t)MALLOC(sizeof (dummy_gss_ctx_id_rec));
376         ctx->token_number = MAGIC_TOKEN_NUMBER;
377         ctx->established = 1;

379         *context_handle = (gss_ctx_id_t)ctx;

381         dprintf("Leaving import_sec_context\n");
382         return (GSS_S_COMPLETE);
383 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
     232 Thu Jul 11 01:30:00 2013
new/usr/src/uts/common/gssapi/mechs/krb5/Makefile
first pass
**********************************************************
```

```
#
# Copyright 1997-2003 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# ident "%Z%%M% %I%     %E% SMI"
#
# /usr/src/uts/common/gssapi/mechs/krb5/Makefile

include ../../../../../Makefile.master

# EXPORT DELETE START

all:
        @$(ECHO) " This Makefile is used to clean up the source tree\n" \
                "for export distribution.\n" \
                "[Usage]: make [EXPORT_SRC] [CRYPT_SRC]\n\n" \
                "WARNING: EXPORT_SRC, CRYPT_SRC targets change the\n" \
                "source tree and remove the Makefile."

# Special target to clean up the source tree for export distribution
# Warning: This target changes the source tree
EXPORT_SRC:
        $(RM) krb5mech.c+ include/gssapiP_krb5.h+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < include/gssapiP_krb5.h > include/gssapiP_krb5.h+
        $(MV) include/gssapiP_krb5.h+ include/gssapiP_krb5.h
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < krb5mech.c > krb5mech.c+
        $(MV) krb5mech.c+ krb5mech.c

        $(RM) crypto/des/f_cbc.c+ crypto/des/f_cksum.c+ \
        crypto/des/d3_cbc.c+ mech/seal.c+ mech/unseal.c+

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < mech/seal.c > mech/seal.c+
        $(MV) mech/seal.c+ mech/seal.c

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < mech/unseal.c > mech/unseal.c+
        $(MV) mech/unseal.c+ mech/unseal.c

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < crypto/des/f_cbc.c > crypto/des/f_cbc.c+
        $(MV) crypto/des/f_cbc.c+ crypto/des/f_cbc.c

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < crypto/des/d3_cbc.c > crypto/des/d3_cbc.c+
        $(MV) crypto/des/d3_cbc.c+ crypto/des/d3_cbc.c

        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < crypto/des/f_cksum.c > crypto/des/f_cksum.c+
        $(MV) crypto/des/f_cksum.c+ crypto/des/f_cksum.c

        $(RM) ../../../Makefile.files+
        sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
                < ../../../Makefile.files > ../../../Makefile.files+
        $(MV) ../../../Makefile.files+ ../../../Makefile.files

        $(RM) Makefile+
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
```



```
        $(MV) Makefile+ Makefile

        $(CHMOD) 444 krb5mech.c include/gssapiP_krb5.h crypto/des/f_cbc.c \
                crypto/des/f_cksum.c crypto/des/d3_cbc.c \
                mech/seal.c mech/unseal.c

# CRYPT DELETE START
# Special target to clean up the source tree for domestic distribution
# Warning: This target changes the source tree
CRYPT_SRC:
        $(RM) krb5mech.c+
        sed -e "/CRYPT DELETE START/,/CRYPT DELETE END/d" \
                < krb5mech.c > krb5mech.c+
        $(MV) krb5mech.c+ krb5mech.c

        $(RM)  Makefile+
        sed -e "/^# CRYPT DELETE START/,/^# CRYPT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile

        $(CHMOD) 444 krb5mech.c Makefile
# CRYPT DELETE END
# EXPORT DELETE END
```

```
**********************************************************
    4049 Thu Jul 11 01:30:01 2013
new/usr/src/uts/common/gssapi/mechs/krb5/crypto/des/d3_cbc.c
first pass
**********************************************************
```

```
/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 1995 by Richard P. Basch.  All Rights Reserved.
 * Copyright 1995 by Lehman Brothers, Inc.  All Rights Reserved.
 *
 * Export of this software from the United States of America may
 *   require a specific license from the United States Government.
 *   It is the responsibility of any person or organization contemplating
 *   export to obtain such a license before exporting.
 *
 * WITHIN THAT CONSTRAINT, permission to use, copy, modify, and
 * distribute this software and its documentation for any purpose and
 * without fee is hereby granted, provided that the above copyright
 * notice appear in all copies and that both that copyright notice and
 * this permission notice appear in supporting documentation, and that
 * the name of Richard P. Basch, Lehman Brothers and M.I.T. not be used
 * in advertising or publicity pertaining to distribution of the software
 * without specific, written prior permission.  Richard P. Basch,
 * Lehman Brothers and M.I.T. make no representations about the suitability
 * of this software for any purpose.  It is provided "as is" without
 * express or implied warranty.
 */

#include "des_int.h"

/*
 * Triple-DES CBC encryption mode.
 */
#ifndef _KERNEL
int
mit_des3_cbc_encrypt(krb5_context context, const mit_des_cblock *in, mit_des_cbl
                     unsigned long length, krb5_keyblock *key,
                     const mit_des_cblock ivec, int encrypt)
{
    int ret = KRB5_PROG_ETYPE_NOSUPP;
/* EXPORT DELETE START */
    KRB5_MECH_TO_PKCS algos;
    CK_MECHANISM mechanism;
    CK_RV rv;
    /* For the Key Object */
    ret = 0;

    if ((rv = get_algo(key->enctype, &algos)) != CKR_OK) {
        KRB5_LOG0(KRB5_ERR, "failure to get algo id in function "
            "mit_des3_cbc_encrypt.");
        ret = PKCS_ERR;
        goto cleanup;
    }

    rv = init_key_uef(krb_ctx_hSession(context), key);
    if (rv != CKR_OK) {
        KRB5_LOG(KRB5_ERR, "init_key_uef failed in "
            "mit_des3_cbc_encrypt: rv = 0x%0x", rv);
        ret = PKCS_ERR;
        goto cleanup;
    }
```



```
    mechanism.mechanism = algos.enc_algo;
    mechanism.pParameter = (void*)ivec;
    if (ivec != NULL)
        mechanism.ulParameterLen = sizeof(mit_des_cblock);
    else
        mechanism.ulParameterLen = 0;

    if (encrypt)
        rv = C_EncryptInit(krb_ctx_hSession(context), &mechanism, key->hKey);
    else
        rv = C_DecryptInit(krb_ctx_hSession(context), &mechanism, key->hKey);

    if (rv != CKR_OK) {
        KRB5_LOG(KRB5_ERR, "C_EncryptInit/C_DecryptInit failed in "
                "mit_des3_cbc_encrypt: rv = 0x%x", rv);
        ret = PKCS_ERR;
        goto cleanup;
    }

    if (encrypt)
        rv = C_Encrypt(krb_ctx_hSession(context), (CK_BYTE_PTR)in,
            (CK_ULONG)length, (CK_BYTE_PTR)out,
            (CK_ULONG_PTR)&length);
    else
        rv = C_Decrypt(krb_ctx_hSession(context), (CK_BYTE_PTR)in,
            (CK_ULONG)length, (CK_BYTE_PTR)out,
            (CK_ULONG_PTR)&length);

    if (rv != CKR_OK) {
            KRB5_LOG(KRB5_ERR,
                "C_Encrypt/C_Decrypt failed in mit_des3_cbc_encrypt: "
                "rv = 0x%x", rv);
            ret = PKCS_ERR;
    }
cleanup:

final_cleanup:
    if (ret)
        (void) memset(out, 0, length);

/* EXPORT DELETE END */
    KRB5_LOG(KRB5_INFO, "mit_des3_cbc_encrypt() end ret=%d\n", ret);
    return(ret);
}

#else
#include <sys/crypto/api.h>

/* ARGSUSED */
int
mit_des3_cbc_encrypt(krb5_context context,
        const mit_des_cblock *in,
        mit_des_cblock *out,
        unsigned long length, krb5_keyblock *key,
        const mit_des_cblock ivec, int encrypt)
{
        int ret = KRB5_PROG_ETYPE_NOSUPP;
/* EXPORT DELETE START */
        krb5_data ivdata;

        KRB5_LOG(KRB5_INFO, "mit_des3_cbc_encrypt() start encrypt=%d", encrypt);

        ivdata.data = (char *)ivec;
        ivdata.length = sizeof(mit_des_cblock);

        ret = k5_ef_crypto((const char *)in, (char *)out,
```

```
 125                         length, key, &ivdata, encrypt);

 130 /* EXPORT DELETE END */
 127         KRB5_LOG(KRB5_INFO, "mit_des3_cbc_encrypt() end retval=%d", ret);
 128         return(ret);
 129 }
```

unchanged_portion_omitted_

```
**********************************************************
    4134 Thu Jul 11 01:30:01 2013
new/usr/src/uts/common/gssapi/mechs/krb5/crypto/des/f_cbc.c
first pass
**********************************************************
/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */


/*
 * Copyright (c) 1990 Dennis Ferguson.  All rights reserved.
 *
 * Commercial use is permitted only if products which are derived from
 * or include this software are made available for purchase and/or use
 * in Canada.  Otherwise, redistribution and use in source and binary
 * forms are permitted.
 */

/*
 * des_cbc_encrypt.c - an implementation of the DES cipher function in cbc mode
 */
#include "des_int.h"

/*
 * des_cbc_encrypt - {en,de}crypt a stream in CBC mode
 */

/* SUNW14resync - sparcv9 cc complained about lack of object init */
/* = all zero */
const mit_des_cblock mit_des_zeroblock = {0, 0, 0, 0, 0, 0, 0, 0};

#undef mit_des_cbc_encrypt

#ifndef _KERNEL
int
mit_des_cbc_encrypt(context, in, out, length, key, ivec, encrypt)
        krb5_context context;
        const mit_des_cblock  *in;
        mit_des_cblock  *out;
        long length;
        krb5_keyblock *key;
        mit_des_cblock ivec;
        int encrypt;
{
    krb5_error_code ret = KRB5_PROG_ETYPE_NOSUPP;
/* EXPORT DELETE START */
    KRB5_MECH_TO_PKCS algos;
    CK_MECHANISM mechanism;
    CK_RV rv;
    /* For the Key Object */

    ret = 0;
    if ((rv = get_algo(key->enctype, &algos)) != CKR_OK) {
        KRB5_LOG0(KRB5_ERR, "failure to get algo id in function "
            "mit_des_cbc_encrypt.");
        ret = PKCS_ERR;
        goto cleanup;
    }

    rv = init_key_uef(krb_ctx_hSession(context), key);
    if (rv != CKR_OK) {
        KRB5_LOG(KRB5_ERR, "init_key_uef failed in "
                "mit_des_cbc_encrypt: rv = 0x%x", rv);
        ret = PKCS_ERR;
        goto cleanup;
    }

    mechanism.mechanism = algos.enc_algo;
    mechanism.pParameter = ivec;
    if (ivec != NULL)
        mechanism.ulParameterLen = MIT_DES_BLOCK_LENGTH;
    else
        mechanism.ulParameterLen = 0;

    if (encrypt)
        rv = C_EncryptInit(krb_ctx_hSession(context), &mechanism, key->hKey);
    else
        rv = C_DecryptInit(krb_ctx_hSession(context), &mechanism, key->hKey);

    if (rv != CKR_OK) {
        KRB5_LOG(KRB5_ERR, "C_EncryptInit/C_DecryptInit failed in "
                "mit_des_cbc_encrypt: rv = 0x%x", rv);
        ret = PKCS_ERR;
        goto cleanup;
    }

    if (encrypt)
        rv = C_Encrypt(krb_ctx_hSession(context), (CK_BYTE_PTR)in,
            (CK_ULONG)length, (CK_BYTE_PTR)out,
            (CK_ULONG_PTR)&length);
    else
        rv = C_Decrypt(krb_ctx_hSession(context), (CK_BYTE_PTR)in,
            (CK_ULONG)length, (CK_BYTE_PTR)out,
            (CK_ULONG_PTR)&length);

    if (rv != CKR_OK) {
            KRB5_LOG(KRB5_ERR,
                "C_Encrypt/C_Decrypt failed in mit_des_cbc_encrypt: "
                "rv = 0x%x", rv);
            ret = PKCS_ERR;
    }
cleanup:

final_cleanup:
    if (ret)
        (void) memset(out, 0, length);

/* EXPORT DELETE END */
    KRB5_LOG(KRB5_INFO, "mit_des_cbc_encrypt() end retval=%d", ret);

    return(ret);
}
#else

/*
 * This routine performs DES cipher-block-chaining operation, either
 * encrypting from cleartext to ciphertext, if encrypt != 0 or
 * decrypting from ciphertext to cleartext, if encrypt == 0.
 *
 * The key schedule is passed as an arg, as well as the cleartext or
 * ciphertext.  The cleartext and ciphertext should be in host order.
 *
 * NOTE-- the output is ALWAYS an multiple of 8 bytes long.  If not
 * enough space was provided, your program will get trashed.
 *
 * For encryption, the cleartext string is null padded, at the end, to
 * an integral multiple of eight bytes.
 *
 * For decryption, the ciphertext will be used in integral multiples
 * of 8 bytes, but only the first "length" bytes returned into the
```

```
126  * cleartext.
127  */

129 /* ARGSUSED */
130 int
131 mit_des_cbc_encrypt(krb5_context context,
132         const mit_des_cblock *in,
133         mit_des_cblock *out,
134         long length, krb5_keyblock *key,
135         mit_des_cblock ivec, int encrypt)
136 {
137         int ret = KRB5_PROG_ETYPE_NOSUPP;
140 /* EXPORT DELETE START */
138         krb5_data ivdata;
139         ret = 0;

141         KRB5_LOG(KRB5_INFO, "mit_des_cbc_encrypt() start encrypt=%d", encrypt);

143         ivdata.data = (char *)ivec;
144         ivdata.length = sizeof(mit_des_cblock);

146         ret = k5_ef_crypto((const char *)in,
147                         (char *)out, length, key, &ivdata, encrypt);

152 /* EXPORT DELETE END */
149         KRB5_LOG(KRB5_INFO, "mit_des_cbc_encrypt() end retval=%d", ret);
150         return(ret);
151 }
```

_____***unchanged_portion_omitted_***

****
1298 Thu Jul 11 01:30:02 2013
new/usr/src/uts/common/gssapi/mechs/krb5/crypto/des/f_cksum.c
first pass
****

```
 1 /*
 2  * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 3  * Use is subject to license terms.
 4  */


 7 /*
 8  * des_cbc_cksum.c - compute an 8 byte checksum using DES in CBC mode
 9  */
10 #include "des_int.h"

12 /*
13  * This routine performs DES cipher-block-chaining checksum operation,
14  * a.k.a.  Message Authentication Code.  It ALWAYS encrypts from input
15  * to a single 64 bit output MAC checksum.
16  *
17  * The key schedule is passed as an arg, as well as the cleartext or
18  * ciphertext. The cleartext and ciphertext should be in host order.
19  *
20  * NOTE-- the output is ALWAYS 8 bytes long.  If not enough space was
21  * provided, your program will get trashed.
22  *
23  * The input is null padded, at the end (highest addr), to an integral
24  * multiple of eight bytes.
25  */
26 unsigned long
27 mit_des_cbc_cksum(krb5_context context,
28         const krb5_octet *in, krb5_octet *out,
29         unsigned long length, krb5_keyblock *key,
30         const krb5_octet  *ivec)
31 {
32         krb5_error_code ret = 0;
33 /* EXPORT DELETE START */
33         krb5_data input;
34         krb5_data output;
35         krb5_data ivecdata;

37         input.data = (char *)in;
38         input.length = length;
39         output.data = (char *)out;
40         output.length = MIT_DES_BLOCK_LENGTH;
41         ivecdata.data = (char *)ivec;
42         ivecdata.length = MIT_DES_BLOCK_LENGTH;

44         ret = k5_ef_mac(context, key, &ivecdata,
45                 (const krb5_data *)&input, &output);

48 /* EXPORT DELETE END */
47         return (ret);
48 }
```

_____***unchanged_portion_omitted_***

****
28510 Thu Jul 11 01:30:02 2013
new/usr/src/uts/common/gssapi/mechs/krb5/include/gssapiP_krb5.h
first pass
****
_____***unchanged_portion_omitted_***

```
 218 extern g_set kg_vdb;

 220 extern k5_mutex_t gssint_krb5_keytab_lock;

 222 /* helper macros */

 224 #define kg_save_name(name)              g_save_name(&kg_vdb,name)
 225 #define kg_save_cred_id(cred)           g_save_cred_id(&kg_vdb,cred)
 226 #define kg_save_ctx_id(ctx)             g_save_ctx_id(&kg_vdb,ctx)
 227 #define kg_save_lucidctx_id(lctx)       g_save_lucidctx_id(&kg_vdb,lctx)

 229 #define kg_validate_name(name)          g_validate_name(&kg_vdb,name)
 230 #define kg_validate_cred_id(cred)       g_validate_cred_id(&kg_vdb,cred)
 231 #define kg_validate_ctx_id(ctx)         g_validate_ctx_id(&kg_vdb,ctx)
 232 #define kg_validate_lucidctx_id(lctx)   g_validate_lucidctx_id(&kg_vdb,lctx)

 234 #define kg_delete_name(name)            g_delete_name(&kg_vdb,name)
 235 #define kg_delete_cred_id(cred)         g_delete_cred_id(&kg_vdb,cred)
 236 #define kg_delete_ctx_id(ctx)           g_delete_ctx_id(&kg_vdb,ctx)
 237 #define kg_delete_lucidctx_id(lctx)     g_delete_lucidctx_id(&kg_vdb,lctx)

 239 /** helper functions **/

 241 OM_uint32 kg_get_defcred
 242         (OM_uint32 *minor_status,
 243                     gss_cred_id_t *cred);

 245 krb5_error_code kg_checksum_channel_bindings
 246          (krb5_context context, gss_channel_bindings_t cb,
 247                                           krb5_checksum *cksum,
 248                                           int bigend);

 250 krb5_error_code kg_make_seq_num (krb5_context context,
 251                                          krb5_keyblock *key,
 252             int direction, krb5_ui_4 seqnum, unsigned char *cksum,
 253                                  unsigned char *buf);

 255 krb5_error_code kg_get_seq_num (krb5_context context,
 256                                          krb5_keyblock *key,
 257             unsigned char *cksum, unsigned char *buf, int *direction,
 258                                        krb5_ui_4 *seqnum);

 260 krb5_error_code kg_make_seed (krb5_context context,
 261                                       krb5_keyblock *key,
 262                                       unsigned char *seed);

 264 int kg_confounder_size (krb5_context context, krb5_keyblock *key);

 266 krb5_error_code kg_make_confounder (krb5_context context,
 267             krb5_keyblock *key, unsigned char *buf);

 269 krb5_error_code kg_encrypt (krb5_context context,
 270                                    krb5_keyblock *key, int usage,
 271                                    krb5_pointer iv,
 272                                    krb5_const_pointer in,
 273                                    krb5_pointer out,
 274                                    unsigned int length);
 275 krb5_error_code
 276 kg_arcfour_docrypt (krb5_context,
```



```
 277                     const krb5_keyblock *longterm_key , int ms_usage,
 278                     const unsigned char *kd_data, size_t kd_data_len,
 279                     const unsigned char *input_buf, size_t input_len,
 280                     unsigned char *output_buf);

 282 krb5_error_code kg_decrypt (krb5_context context,
 283                                       krb5_keyblock *key,  int usage,
 284                                       krb5_pointer iv,
 285                                       krb5_const_pointer in,
 286                                       krb5_pointer out,
 287                                       unsigned int length);

 289 OM_uint32 kg_seal (OM_uint32 *minor_status,
 290                    gss_ctx_id_t context_handle,
 291                    int conf_req_flag,
 292                    int qop_req,
 293                    gss_buffer_t input_message_buffer,
 294                    int *conf_state,
 295                    gss_buffer_t output_message_buffer,
 296                    int toktype);

 298 OM_uint32 kg_unseal (OM_uint32 *minor_status,
 299                      gss_ctx_id_t context_handle,
 300                      gss_buffer_t input_token_buffer,
 301                      gss_buffer_t message_buffer,
 302                      int *conf_state,
 303                      int *qop_state,
 304                      int toktype);

 306 OM_uint32 kg_seal_size (OM_uint32 *minor_status,
 307                                    gss_ctx_id_t context_handle,
 308                                    int conf_req_flag,
 309                                    gss_qop_t qop_req,
 310                                    OM_uint32 output_size,
 311                                    OM_uint32 *input_size);

 313 krb5_error_code kg_ctx_size (krb5_context kcontext,
 314                                        krb5_pointer arg,
 315                                        size_t *sizep);

 317 krb5_error_code kg_ctx_externalize (krb5_context kcontext,
 318                                             krb5_pointer arg,
 319                                             krb5_octet **buffer,
 320                                             size_t *lenremain);

 322 krb5_error_code kg_ctx_internalize (krb5_context kcontext,
 323                                             krb5_pointer *argp,
 324                                             krb5_octet **buffer,
 325                                             size_t *lenremain);

 327 OM_uint32 kg_sync_ccache_name (krb5_context context, OM_uint32 *minor_status);

 329 OM_uint32 kg_caller_provided_ccache_name (OM_uint32 *minor_status,
 330                                           int *out_caller_provided_name);

 332 OM_uint32 kg_get_ccache_name (OM_uint32 *minor_status,
 333                                const char **out_name);

 335 OM_uint32 kg_set_ccache_name (OM_uint32 *minor_status,
 336                                const char *name);

 338 /** declarations of internal name mechanism functions **/

 340 OM_uint32 krb5_gss_acquire_cred
 341 (OM_uint32*,       /* minor_status */
 342             gss_name_t,       /* desired_name */
```

```
343             OM_uint32,        /* time_req */
344             gss_OID_set,      /* desired_mechs */
345             gss_cred_usage_t, /* cred_usage */
346             gss_cred_id_t*,   /* output_cred_handle */
347             gss_OID_set*,     /* actual_mechs */
348             OM_uint32*        /* time_rec */
349            );

351 OM_uint32 krb5_gss_release_cred
352 (OM_uint32*,       /* minor_status */
353             gss_cred_id_t*    /* cred_handle */
354            );

356 OM_uint32 krb5_gss_init_sec_context
357 (OM_uint32*,       /* minor_status */
358             gss_cred_id_t,    /* claimant_cred_handle */
359             gss_ctx_id_t*,    /* context_handle */
360             gss_name_t,       /* target_name */
361             gss_OID,          /* mech_type */
362             OM_uint32,        /* req_flags */
363             OM_uint32,        /* time_req */
364             gss_channel_bindings_t,
365                               /* input_chan_bindings */
366             gss_buffer_t,     /* input_token */
367             gss_OID*,         /* actual_mech_type */
368             gss_buffer_t,     /* output_token */
369             OM_uint32*,       /* ret_flags */
370             OM_uint32*        /* time_rec */
371            );

373 OM_uint32 krb5_gss_accept_sec_context
374 (OM_uint32*,       /* minor_status */
375             gss_ctx_id_t*,    /* context_handle */
376             gss_cred_id_t,    /* verifier_cred_handle */
377             gss_buffer_t,     /* input_token_buffer */
378             gss_channel_bindings_t,
379                               /* input_chan_bindings */
380             gss_name_t*,      /* src_name */
381             gss_OID*,         /* mech_type */
382             gss_buffer_t,     /* output_token */
383             OM_uint32*,       /* ret_flags */
384             OM_uint32*,       /* time_rec */
385             gss_cred_id_t*    /* delegated_cred_handle */
386            );

388 OM_uint32 krb5_gss_process_context_token
389 (OM_uint32*,       /* minor_status */
390             gss_ctx_id_t,     /* context_handle */
391             gss_buffer_t      /* token_buffer */
392            );

394 OM_uint32 krb5_gss_delete_sec_context
395 (OM_uint32*,       /* minor_status */
396             gss_ctx_id_t*,    /* context_handle */
397             gss_buffer_t      /* output_token */
398 #ifdef  _KERNEL
399             /* */, OM_uint32    /* context verifier */
400 #endif
401            );

403 OM_uint32 krb5_gss_context_time
404 (OM_uint32*,       /* minor_status */
405             gss_ctx_id_t,     /* context_handle */
406             OM_uint32*        /* time_rec */
407            );
```



```
409 OM_uint32 krb5_gss_sign
410 (OM_uint32*,       /* minor_status */
411             gss_ctx_id_t,     /* context_handle */
412             int,              /* qop_req */
413             gss_buffer_t,     /* message_buffer */
414             gss_buffer_t      /* message_token */
415 #ifdef  _KERNEL
416             /* */, OM_uint32    /* context verifier */
417 #endif
418            );

420 OM_uint32 krb5_gss_verify
421 (OM_uint32*,       /* minor_status */
422             gss_ctx_id_t,     /* context_handle */
423             gss_buffer_t,     /* message_buffer */
424             gss_buffer_t,     /* token_buffer */
425             int*              /* qop_state */
426 #ifdef  _KERNEL
427             /* */, OM_uint32    /* context verifier */
428 #endif
429            );

431 /* EXPORT DELETE START */
431 OM_uint32 krb5_gss_seal
432 (OM_uint32*,       /* minor_status */
433             gss_ctx_id_t,     /* context_handle */
434             int,              /* conf_req_flag */
435             int,              /* qop_req */
436             gss_buffer_t,     /* input_message_buffer */
437             int*,             /* conf_state */
438             gss_buffer_t      /* output_message_buffer */
439 #ifdef  _KERNEL
440             /* */, OM_uint32    /* context verifier */
441 #endif
442            );

444 OM_uint32 krb5_gss_unseal
445 (OM_uint32*,       /* minor_status */
446             gss_ctx_id_t,     /* context_handle */
447             gss_buffer_t,     /* input_message_buffer */
448             gss_buffer_t,     /* output_message_buffer */
449             int*,             /* conf_state */
450             int*              /* qop_state */
451 #ifdef  _KERNEL
452             /* */, OM_uint32    /* context verifier */
453 #endif
454            );
456 /* EXPORT DELETE END */

456 OM_uint32 krb5_gss_display_status
457 (OM_uint32*,       /* minor_status */
458             OM_uint32,        /* status_value */
459             int,              /* status_type */
460             gss_OID,          /* mech_type */
461             OM_uint32*,       /* message_context */
462             gss_buffer_t      /* status_string */
463            );

465 OM_uint32 krb5_gss_indicate_mechs
466 (OM_uint32*,       /* minor_status */
467             gss_OID_set*      /* mech_set */
468            );

470 OM_uint32 krb5_gss_compare_name
471 (OM_uint32*,       /* minor_status */
472             gss_name_t,       /* name1 */
```

```
473             gss_name_t,       /* name2 */
474             int*              /* name_equal */
475           );

477 OM_uint32 krb5_gss_display_name
478 (OM_uint32*,       /* minor_status */
479             gss_name_t,       /* input_name */
480             gss_buffer_t,     /* output_name_buffer */
481             gss_OID*          /* output_name_type */
482           );

484 OM_uint32 krb5_gss_import_name
485 (OM_uint32*,       /* minor_status */
486             gss_buffer_t,     /* input_name_buffer */
487             gss_OID,          /* input_name_type */
488             gss_name_t*       /* output_name */
489           );

491 OM_uint32 krb5_gss_release_name
492 (OM_uint32*,       /* minor_status */
493             gss_name_t*       /* input_name */
494           );

496 OM_uint32 krb5_gss_inquire_cred
497 (OM_uint32 *,      /* minor_status */
498             gss_cred_id_t,    /* cred_handle */
499             gss_name_t *,     /* name */
500             OM_uint32 *,      /* lifetime */
501             gss_cred_usage_t*,/* cred_usage */
502             gss_OID_set *     /* mechanisms */
503           );

505 OM_uint32 krb5_gss_inquire_context
506 (OM_uint32*,       /* minor_status */
507             gss_ctx_id_t,     /* context_handle */
508             gss_name_t*,      /* initiator_name */
509             gss_name_t*,      /* acceptor_name */
510             OM_uint32*,       /* lifetime_rec */
511             gss_OID*,         /* mech_type */
512             OM_uint32*,       /* ret_flags */
513             int*,             /* locally_initiated */
514             int*              /* open */
515           );

517 /* New V2 entry points */
518 OM_uint32 krb5_gss_get_mic
519 (OM_uint32 *,           /* minor_status */
520             gss_ctx_id_t,               /* context_handle */
521             gss_qop_t,                  /* qop_req */
522             gss_buffer_t,               /* message_buffer */
523             gss_buffer_t                /* message_token */
524           );

526 OM_uint32 krb5_gss_verify_mic
527 (OM_uint32 *,           /* minor_status */
528             gss_ctx_id_t,               /* context_handle */
529             gss_buffer_t,               /* message_buffer */
530             gss_buffer_t,               /* message_token */
531             gss_qop_t *                 /* qop_state */
532           );

534 OM_uint32 krb5_gss_wrap
535 (OM_uint32 *,      /* minor_status */
536             gss_ctx_id_t,               /* context_handle */
537             int,                        /* conf_req_flag */
538             gss_qop_t,                  /* qop_req */
```



```
539             gss_buffer_t,               /* input_message_buffer */
540             int *,                      /* conf_state */
541             gss_buffer_t                /* output_message_buffer */
542           );

544 OM_uint32 krb5_gss_unwrap
545 (OM_uint32 *,           /* minor_status */
546             gss_ctx_id_t,               /* context_handle */
547             gss_buffer_t,               /* input_message_buffer */
548             gss_buffer_t,               /* output_message_buffer */
549             int *,                      /* conf_state */
550             gss_qop_t *                 /* qop_state */
551           );

553 OM_uint32 krb5_gss_wrap_size_limit
554 (OM_uint32 *,           /* minor_status */
555             gss_ctx_id_t,               /* context_handle */
556             int,                        /* conf_req_flag */
557             gss_qop_t,                  /* qop_req */
558             OM_uint32,                  /* req_output_size */
559             OM_uint32 *                 /* max_input_size */
560           );

562 OM_uint32 krb5_gss_import_name_object
563 (OM_uint32 *,           /* minor_status */
564             void *,                     /* input_name */
565             gss_OID,                    /* input_name_type */
566             gss_name_t *                /* output_name */
567           );

569 OM_uint32 krb5_gss_export_name_object
570 (OM_uint32 *,           /* minor_status */
571             gss_name_t,                 /* input_name */
572             gss_OID,                    /* desired_name_type */
573             void * *                    /* output_name */
574           );

576 OM_uint32 krb5_gss_add_cred
577 (OM_uint32 *,           /* minor_status */
578             gss_cred_id_t,              /* input_cred_handle */
579             gss_name_t,                 /* desired_name */
580             gss_OID,                    /* desired_mech */
581             gss_cred_usage_t,           /* cred_usage */
582             OM_uint32,                  /* initiator_time_req */
583             OM_uint32,                  /* acceptor_time_req */
584             gss_cred_id_t *,            /* output_cred_handle */
585             gss_OID_set *,              /* actual_mechs */
586             OM_uint32 *,                /* initiator_time_rec */
587             OM_uint32 *                 /* acceptor_time_rec */
588           );

590 OM_uint32 krb5_gss_inquire_cred_by_mech
591 (OM_uint32  *,          /* minor_status */
592             gss_cred_id_t,              /* cred_handle */
593             gss_OID,                    /* mech_type */
594             gss_name_t *,               /* name */
595             OM_uint32 *,                /* initiator_lifetime */
596             OM_uint32 *,                /* acceptor_lifetime */
597             gss_cred_usage_t *          /* cred_usage */
598           );

600 OM_uint32 krb5_gss_export_sec_context
601 (OM_uint32 *,           /* minor_status */
602             gss_ctx_id_t *,             /* context_handle */
603             gss_buffer_t                /* interprocess_token */
604             );
```

```
606 OM_uint32 krb5_gss_import_sec_context
607 (OM_uint32 *,               /* minor_status */
608             gss_buffer_t,              /* interprocess_token */
609             gss_ctx_id_t *             /* context_handle */
610             /* Note no _KERNEL context verifier */
611             );

613 krb5_error_code krb5_gss_ser_init(krb5_context);

615 OM_uint32 krb5_gss_release_oid
616 (OM_uint32 *,               /* minor_status */
617             gss_OID *                  /* oid */
618            );

620 OM_uint32 krb5_gss_internal_release_oid
621 (OM_uint32 *,               /* minor_status */
622             gss_OID *                  /* oid */
623            );

625 OM_uint32 krb5_gss_inquire_names_for_mech
626 (OM_uint32 *,               /* minor_status */
627             gss_OID,                   /* mechanism */
628             gss_OID_set *              /* name_types */
629            );

631 /* SUNW15resync - XXX nullify? */
632 OM_uint32 krb5_gss_canonicalize_name
633 (OM_uint32  *,              /* minor_status */
634             const gss_name_t,          /* input_name */
635             const gss_OID,             /* mech_type */
636             gss_name_t *               /* output_name */
637          );
638
639 OM_uint32 krb5_gss_export_name
640 (OM_uint32  *,              /* minor_status */
641             const gss_name_t,          /* input_name */
642             gss_buffer_t               /* exported_name */
643          );

645 OM_uint32 krb5_gss_duplicate_name
646 (OM_uint32  *,              /* minor_status */
647             const gss_name_t,          /* input_name */
648             gss_name_t *               /* dest_name */
649          );

651 OM_uint32 krb5_gss_validate_cred
652 (OM_uint32 *,               /* minor_status */
653             gss_cred_id_t              /* cred */
654          );

656 OM_uint32
657 krb5_gss_validate_cred_1(OM_uint32 * /* minor_status */,
658                          gss_cred_id_t /* cred_handle */,
659                          krb5_context /* context */);

661 gss_OID krb5_gss_convert_static_mech_oid(gss_OID oid);
662
663 krb5_error_code gss_krb5int_make_seal_token_v3(krb5_context,
664                                                krb5_gss_ctx_id_rec *,
665                                                const gss_buffer_desc *,
666                                                gss_buffer_t,
667                                                int, int);

669 OM_uint32 gss_krb5int_unseal_token_v3(krb5_context *contextptr,
670                                       OM_uint32 *minor_status,
```



```
671                                       krb5_gss_ctx_id_rec *ctx,
672                                       unsigned char *ptr, int bodysize,
673                                       gss_buffer_t message_buffer,
674                                       int *conf_state, int *qop_state,
675                                       int toktype);

677 /*
678  * SUNW15resync
679  * Solaris specific interfaces start
680  */

682 OM_uint32 krb5_gss_store_cred (
683             OM_uint32 *,           /* minor_status */
684             const gss_cred_id_t,   /* input_cred */
685             gss_cred_usage_t,      /* cred_usage */
686             const gss_OID,         /* desired_mech */
687             OM_uint32,             /* overwrite_cred */
688             OM_uint32,             /* default_cred */
689             gss_OID_set *,         /* elements_stored */
690             gss_cred_usage_t *     /* cred_usage_stored */
691            );

693 OM_uint32 krb5_pname_to_uid(
694                 OM_uint32 *,           /* minor status */
695                 const gss_name_t,      /* pname */
696                 uid_t *                /* uidOUt */
697                 );

699 OM_uint32 krb5_gss_userok(
700         OM_uint32 *,            /* minor status */
701         const gss_name_t,       /* remote user principal name */
702         const char *,           /* local unix user name */
703         int *                   /* remote user ok to login w/out pw? */
704         );


707 /*
708  * SUNW15resync
709  * Solaris specific interfaces end
710  */


713 /*
714  * These take unglued krb5-mech-specific contexts.
715  */

717 #define GSS_KRB5_GET_TKT_FLAGS_OID_LENGTH 11
718 #define GSS_KRB5_GET_TKT_FLAGS_OID "\x2a\x86\x48\x86\xf7\x12\x01\x02\x02\x05\x01

720 #ifndef _KERNEL
721 OM_uint32  gss_krb5int_get_tkt_flags
722 (OM_uint32 *minor_status,
723  const gss_ctx_id_t context_handle,
724  const gss_OID desired_object,
725  gss_buffer_set_t *data_set);


728 OM_uint32 KRB5_CALLCONV gss_krb5int_copy_ccache
729         (OM_uint32 *minor_status,
730                  gss_cred_id_t cred_handle,
731                  krb5_ccache out_ccache);

733 #define GSS_KRB5_SET_ALLOWABLE_ENCTYPES_OID_LENGTH 11
734 #define GSS_KRB5_SET_ALLOWABLE_ENCTYPES_OID "\x2a\x86\x48\x86\xf7\x12\x01\x02\x0

736 struct krb5_gss_set_allowable_enctypes_req {
```

```
 737     OM_uint32 num_ktypes;
 738     krb5_enctype *ktypes;
 739 };
```

*unchanged_portion_omitted_*

```
**********************************************************
    6946 Thu Jul 11 01:30:03 2013
new/usr/src/uts/common/gssapi/mechs/krb5/krb5mech.c
first pass
**********************************************************
```

```
/*
 * 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 2007 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 * Copyright (c) 2011 Bayard G. Bell. All rights reserved.
 *
 * A module for Kerberos V5  security mechanism.
 *
 */

#include <sys/types.h>
#include <sys/modctl.h>
#include <sys/errno.h>
#include <mechglueP.h>
#include <gssapiP_krb5.h>
#include <gssapi_err_generic.h>
#include <gssapi/kgssapi_defs.h>
#include <sys/debug.h>
#include <k5-int.h>

/* mechglue wrappers */

static OM_uint32 k5glue_delete_sec_context
        (void *, OM_uint32 *,   /* minor_status */
        gss_ctx_id_t *, /* context_handle */
        gss_buffer_t,   /* output_token */
        OM_uint32);

static OM_uint32 k5glue_sign
        (void *, OM_uint32 *,   /* minor_status */
        gss_ctx_id_t,   /* context_handle */
        int,            /* qop_req */
        gss_buffer_t,   /* message_buffer */
        gss_buffer_t,   /* message_token */
        OM_uint32);

static OM_uint32 k5glue_verify
        (void *, OM_uint32 *,   /* minor_status */
        gss_ctx_id_t,   /* context_handle */
        gss_buffer_t,   /* message_buffer */
        gss_buffer_t,   /* token_buffer */
        int *,  /* qop_state */
```



```
        OM_uint32);

/* EXPORT DELETE START */
static OM_uint32 k5glue_seal
        (void *, OM_uint32 *,   /* minor_status */
        gss_ctx_id_t,           /* context_handle */
        int,                    /* conf_req_flag */
        int,                    /* qop_req */
        gss_buffer_t,           /* input_message_buffer */
        int *,                  /* conf_state */
        gss_buffer_t,           /* output_message_buffer */
        OM_uint32);

static OM_uint32 k5glue_unseal
        (void *, OM_uint32 *,   /* minor_status */
        gss_ctx_id_t,           /* context_handle */
        gss_buffer_t,           /* input_message_buffer */
        gss_buffer_t,           /* output_message_buffer */
        int *,                  /* conf_state */
        int *,                  /* qop_state */
        OM_uint32);
/* EXPORT DELETE END */

static OM_uint32 k5glue_import_sec_context
        (void *, OM_uint32 *,           /* minor_status */
        gss_buffer_t,                   /* interprocess_token */
        gss_ctx_id_t *);                /* context_handle */



static  struct  gss_config krb5_mechanism =
        {{9, "\052\206\110\206\367\022\001\002\002"},
        NULL,   /* context */
        NULL,   /* next */
        TRUE,   /* uses_kmod */
/* EXPORT DELETE START */ /* CRYPT DELETE START */
        k5glue_unseal,
/* EXPORT DELETE END */ /* CRYPT DELETE END */
        k5glue_delete_sec_context,
/* EXPORT DELETE START */ /* CRYPT DELETE START */
        k5glue_seal,
/* EXPORT DELETE END */ /* CRYPT DELETE END */
        k5glue_import_sec_context,
/* EXPORT DELETE START */
/* CRYPT DELETE START */
#if 0
/* CRYPT DELETE END */
        k5glue_seal,
        k5glue_unseal,
/* CRYPT DELETE START */
#endif
/* CRYPT DELETE END */
/* EXPORT DELETE END */
        k5glue_sign,
        k5glue_verify,
        };

static gss_mechanism
        gss_mech_initialize()
{
        return (&krb5_mechanism);
}
______unchanged_portion_omitted_

/* EXPORT DELETE START */
/* V1 only */
```

```
/* ARGSUSED */
static OM_uint32
k5glue_seal(ctx, minor_status, context_handle, conf_req_flag, qop_req,
            input_message_buffer, conf_state, output_message_buffer,
            gssd_ctx_verifier)
        void *ctx;
        OM_uint32 *minor_status;
        gss_ctx_id_t context_handle;
        int conf_req_flag;
        int qop_req;
        gss_buffer_t input_message_buffer;
        int *conf_state;
        gss_buffer_t output_message_buffer;
        OM_uint32 gssd_ctx_verifier;
{
        return (krb5_gss_seal(minor_status, context_handle,
                        conf_req_flag, qop_req, input_message_buffer,
                        conf_state, output_message_buffer, gssd_ctx_verifier));
}
/* EXPORT DELETE END */

/* ARGSUSED */
static OM_uint32
k5glue_sign(ctx, minor_status, context_handle,
                qop_req, message_buffer,
                message_token, gssd_ctx_verifier)
        void *ctx;
        OM_uint32 *minor_status;
        gss_ctx_id_t context_handle;
        int qop_req;
        gss_buffer_t message_buffer;
        gss_buffer_t message_token;
        OM_uint32 gssd_ctx_verifier;
{
        return (krb5_gss_sign(minor_status, context_handle,
                qop_req, message_buffer, message_token, gssd_ctx_verifier));
}

/* EXPORT DELETE START */
/* ARGSUSED */
static OM_uint32
k5glue_unseal(ctx, minor_status, context_handle, input_message_buffer,
            output_message_buffer, conf_state, qop_state, gssd_ctx_verifier)
        void *ctx;
        OM_uint32 *minor_status;
        gss_ctx_id_t context_handle;
        gss_buffer_t input_message_buffer;
        gss_buffer_t output_message_buffer;
        int *conf_state;
        int *qop_state;
        OM_uint32 gssd_ctx_verifier;
{
        return (krb5_gss_unseal(minor_status, context_handle,
                                input_message_buffer, output_message_buffer,
                                conf_state, qop_state, gssd_ctx_verifier));
}
/* EXPORT DELETE END */

/* V1 only */
/* ARGSUSED */
static OM_uint32
k5glue_verify(ctx, minor_status, context_handle, message_buffer,
            token_buffer, qop_state, gssd_ctx_verifier)
        void *ctx;
        OM_uint32 *minor_status;
        gss_ctx_id_t context_handle;
```



```
        gss_buffer_t message_buffer;
        gss_buffer_t token_buffer;
        int *qop_state;
        OM_uint32 gssd_ctx_verifier;
{
        return (krb5_gss_verify(minor_status,
                                context_handle,
                                message_buffer,
                                token_buffer,
                                qop_state, gssd_ctx_verifier));
}
```

unchanged_portion_omitted_

****
2842 Thu Jul 11 01:30:04 2013
new/usr/src/uts/common/gssapi/mechs/krb5/mech/seal.c
first pass
****

```
  1 /* EXPORT DELETE START */

  1 /*
  2  * Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
  3  * Use is subject to license terms.
  4  */

  6 #pragma ident   "%Z%%M% %I%     %E% SMI"

  8 /*
  9  * Copyright 1993 by OpenVision Technologies, Inc.
 10  *
 11  * Permission to use, copy, modify, distribute, and sell this software
 12  * and its documentation for any purpose is hereby granted without fee,
 13  * provided that the above copyright notice appears in all copies and
 14  * that both that copyright notice and this permission notice appear in
 15  * supporting documentation, and that the name of OpenVision not be used
 16  * in advertising or publicity pertaining to distribution of the software
 17  * without specific, written prior permission. OpenVision makes no
 18  * representations about the suitability of this software for any
 19  * purpose.  It is provided "as is" without express or implied warranty.
 20  *
 21  * OPENVISION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
 22  * INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO
 23  * EVENT SHALL OPENVISION BE LIABLE FOR ANY SPECIAL, INDIRECT OR
 24  * CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF
 25  * USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
 26  * OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
 27  * PERFORMANCE OF THIS SOFTWARE.
 28  */

 30 #include "gssapiP_krb5.h"

 32 /*
 33  * $Id: seal.c 16171 2004-03-15 17:45:01Z raeburn $
 34  */

 36 /*ARGSUSED*/
 37 OM_uint32
 38 krb5_gss_seal(minor_status, context_handle, conf_req_flag,
 39               qop_req, input_message_buffer, conf_state,
 40               output_message_buffer
 41 #ifdef   _KERNEL
 42              , gssd_ctx_verifier
 43 #endif
 44              )
 45      OM_uint32 *minor_status;
 46      gss_ctx_id_t context_handle;
 47      int conf_req_flag;
 48      int qop_req;
 49      gss_buffer_t input_message_buffer;
 50      int *conf_state;
 51      gss_buffer_t output_message_buffer;
 52 #ifdef   _KERNEL
 53      OM_uint32 gssd_ctx_verifier;
 54 #endif
 55 {
 56 #ifdef  KRB5_NO_PRIVACY
 57         /*
 58          * conf_req_flag must be zero;
 59          * encryption is disallowed
```



```
 60          * for global version
 61          */
 62       if (conf_req_flag)
 63         return (GSS_S_FAILURE);
 64 #endif
 65
 66    return(kg_seal(minor_status, context_handle, conf_req_flag,
 67                   qop_req, input_message_buffer, conf_state,
 68                   output_message_buffer, KG_TOK_SEAL_MSG));
 69 }
_____unchanged_portion_omitted_

 95 /* EXPORT DELETE END */
```

```
**********************************************************
    2635 Thu Jul 11 01:30:04 2013
new/usr/src/uts/common/gssapi/mechs/krb5/mech/unseal.c
first pass
**********************************************************
```

```
  1 /* EXPORT DELETE START */
```

```
  1 /*
  2  * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
  3  * Use is subject to license terms.
  4  */


  7 /*
  8  * Copyright 1993 by OpenVision Technologies, Inc.
  9  *
 10  * Permission to use, copy, modify, distribute, and sell this software
 11  * and its documentation for any purpose is hereby granted without fee,
 12  * provided that the above copyright notice appears in all copies and
 13  * that both that copyright notice and this permission notice appear in
 14  * supporting documentation, and that the name of OpenVision not be used
 15  * in advertising or publicity pertaining to distribution of the software
 16  * without specific, written prior permission. OpenVision makes no
 17  * representations about the suitability of this software for any
 18  * purpose.  It is provided "as is" without express or implied warranty.
 19  *
 20  * OPENVISION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
 21  * INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO
 22  * EVENT SHALL OPENVISION BE LIABLE FOR ANY SPECIAL, INDIRECT OR
 23  * CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF
 24  * USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
 25  * OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
 26  * PERFORMANCE OF THIS SOFTWARE.
 27  */

 29 #include "gssapiP_krb5.h"

 31 /*
 32  * $Id: unseal.c 16171 2004-03-15 17:45:01Z raeburn $
 33  */

 35 /*ARGSUSED*/
 36 OM_uint32
 37 krb5_gss_unseal(minor_status, context_handle,
 38                 input_message_buffer, output_message_buffer,
 39                 conf_state, qop_state
 40 #ifdef   _KERNEL
 41                 , gssd_ctx_verifier
 42 #endif
 43                 )
 44      OM_uint32 *minor_status;
 45      gss_ctx_id_t context_handle;
 46      gss_buffer_t input_message_buffer;
 47      gss_buffer_t output_message_buffer;
 48      int *conf_state;
 49      int *qop_state;
 50 #ifdef   _KERNEL
 51      OM_uint32 gssd_ctx_verifier;
 52 #endif
 53 {
 54    return(kg_unseal(minor_status, context_handle,
 55                     input_message_buffer, output_message_buffer,
 56                     conf_state, qop_state, KG_TOK_SEAL_MSG));
 57 }
```

unchanged_portion_omitted_


```
 88 /* EXPORT DELETE END */
```

```
**********************************************************
   75164 Thu Jul 11 01:30:05 2013
new/usr/src/uts/common/io/timod.c
onc plus-be-gone
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /* ONC_PLUS EXTRACT START */
  21 /*
  22  * Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
  23  * Use is subject to license terms.
  24  */
  25 /*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
  26 /*        All Rights Reserved   */


  29 /*
  30  * Transport Interface Library cooperating module - issue 2
  31  */

  34 /* ONC_PLUS EXTRACT END */
  33 #include <sys/param.h>
  34 #include <sys/types.h>
  35 #include <sys/stream.h>
  36 #include <sys/stropts.h>
  37 #include <sys/strsubr.h>
  38 #define _SUN_TPI_VERSION 2
  39 #include <sys/tihdr.h>
  40 #include <sys/timod.h>
  41 #include <sys/suntpi.h>
  42 #include <sys/debug.h>
  43 #include <sys/strlog.h>
  44 #include <sys/errno.h>
  45 #include <sys/cred.h>
  46 #include <sys/cmn_err.h>
  47 #include <sys/kmem.h>
  48 #include <sys/sysmacros.h>
  49 #include <sys/ddi.h>
  50 #include <sys/sunddi.h>
  51 #include <sys/strsun.h>
  52 #include <c2/audit.h>

  54 /*
  55  * This is the loadable module wrapper.
  56  */
  57 #include <sys/conf.h>
  58 #include <sys/modctl.h>
```



```
  60 static struct streamtab timinfo;

  62 static struct fmodsw fsw = {
  63         "timod",
  64         &timinfo,
  65         D_MTQPAIR | D_MP,
  66 };
_____unchanged_portion_omitted_


 170 /*
 171  * Local flags used with tim_flags field in instance structure of
 172  * type 'struct _ti_user' declared above.
 173  * Historical note:
 174  * This namespace constants were previously declared in a
 175  * a very messed up namespace in timod.h
 176  *
 177  * There may be 3 states for transport:
 178  *
 179  * 1) It provides T_CAPABILITY_REQ
 180  * 2) It does not provide T_CAPABILITY_REQ
 181  * 3) It is not known yet whether transport provides T_CAPABILITY_REQ or not.
 182  *
 183  * It is assumed that the underlying transport either provides
 184  * T_CAPABILITY_REQ or not and this does not changes during the
 185  * system lifetime.
 186  *
 187  */
 188 #define PEEK_RDQ_EXPIND 0x0001  /* look for expinds on stream rd queues */
 189 #define WAITIOCACK      0x0002  /* waiting for info for ioctl act       */
 190 #define CLTS            0x0004  /* connectionless transport             */
 191 #define COTS            0x0008  /* connection-oriented transport        */
 192 #define CONNWAIT        0x0010  /* waiting for connect confirmation     */
 193 #define LOCORDREL       0x0020  /* local end has orderly released       */
 194 #define REMORDREL       0x0040  /* remote end had orderly released      */
 195 #define NAMEPROC        0x0080  /* processing a NAME ioctl              */
 198 /* ONC_PLUS EXTRACT START */
 196 #define DO_MYNAME       0x0100  /* timod handles TI_GETMYNAME           */
 200 /* ONC_PLUS EXTRACT END */
 197 #define DO_PEERNAME     0x0200  /* timod handles TI_GETPEERNAME         */
 198 #define TI_CAP_RECVD    0x0400  /* TI_CAPABILITY received               */
 199 #define CAP_WANTS_INFO  0x0800  /* TI_CAPABILITY has TC1_INFO set       */
 200 #define WAIT_IOCINFOACK 0x1000  /* T_INFO_REQ generated from ioctl      */
 201 #define WAIT_CONNRESACK 0x2000  /* waiting for T_OK_ACK to T_CONN_RES   */


 204 /* Debugging facilities */
 205 /*
 206  * Logging needed for debugging timod should only appear in DEBUG kernel.
 207  */
 208 #ifdef DEBUG
 209 #define TILOG(msg, arg)         tilog((msg), (arg))
 210 #define TILOGP(msg, arg)        tilogp((msg), (arg))
 211 #else
 212 #define TILOG(msg, arg)
 213 #define TILOGP(msg, arg)
 214 #endif


 217 /*
 218  * Sleep timeout for T_CAPABILITY_REQ. This message never travels across
 219  * network, so timeout value should be enough to cover all internal processing
 220  * time.
 221  */
 222 clock_t tim_tcap_wait = 2;
```

```
224 /* Sleep timeout in tim_recover() */
225 #define TIMWAIT (1*hz)
226 /* Sleep timeout in tim_ioctl_retry() 0.2 seconds */
227 #define TIMIOCWAIT      (200*hz/1000)

229 /*
230  * Return values for ti_doname().
231  */
232 #define DONAME_FAIL     0       /* failing ioctl (done) */
233 #define DONAME_DONE     1       /* done processing */
234 #define DONAME_CONT     2       /* continue proceesing (not done yet) */

236 /*
237  * Function prototypes
238  */
239 static int ti_doname(queue_t *, mblk_t *);
240 static int ti_expind_on_rdqueues(queue_t *);
241 static void tim_ioctl_send_reply(queue_t *, mblk_t *, mblk_t *);
242 static void tim_send_ioc_error_ack(queue_t *, struct tim_tim *, mblk_t *);
243 static void tim_tcap_timer(void *);
244 static void tim_tcap_genreply(queue_t *, struct tim_tim *);
245 static void tim_send_reply(queue_t *, mblk_t *, struct tim_tim *, t_scalar_t);
246 static void tim_answer_ti_sync(queue_t *, mblk_t *, struct tim_tim *,
247     mblk_t *, uint32_t);
248 static void tim_send_ioctl_tpi_msg(queue_t *, mblk_t *, struct tim_tim *,
249         struct iocblk *);
250 static void tim_clear_peer(struct tim_tim *);

252 int
253 _init(void)
254 {
255         int     error;

257         rw_init(&tim_list_rwlock, NULL, RW_DRIVER, NULL);
258         error = mod_install(&modlinkage);
259         if (error != 0) {
260                 rw_destroy(&tim_list_rwlock);
261                 return (error);
262         }

264         return (0);
265 }
```

*unchanged_portion_omitted_*

```
286 /*
287  * Hash list for all instances. Used to find tim_tim structure based on
288  * ACCEPTOR_id in T_CONN_RES. Protected by tim_list_rwlock.
289  */
290 #define TIM_HASH_SIZE   256
291 #ifdef  _ILP32
292 #define TIM_HASH(id) (((uintptr_t)(id) >> 8) % TIM_HASH_SIZE)
293 #else
294 #define TIM_HASH(id) ((uintptr_t)(id) % TIM_HASH_SIZE)
295 #endif  /* _ILP32 */
296 static struct tim_tim   *tim_hash[TIM_HASH_SIZE];
297 int             tim_cnt = 0;

299 static void tilog(char *, t_scalar_t);
300 static void tilogp(char *, uintptr_t);
301 static mblk_t *tim_filladdr(queue_t *, mblk_t *, boolean_t);
302 static void tim_addlink(struct tim_tim  *);
303 static void tim_dellink(struct tim_tim  *);
304 static struct tim_tim *tim_findlink(t_uscalar_t);
305 static void tim_recover(queue_t *, mblk_t *, t_scalar_t);
306 static void tim_ioctl_retry(queue_t *);
```



```
308 int dotilog = 0;

310 #define TIMOD_ID        3

316 /* ONC_PLUS EXTRACT START */
312 static int timodopen(queue_t *, dev_t *, int, int, cred_t *);
318 /* ONC_PLUS EXTRACT END */
313 static int timodclose(queue_t *, int, cred_t *);
314 static void timodwput(queue_t *, mblk_t *);
315 static void timodrput(queue_t *, mblk_t *);
322 /* ONC_PLUS EXTRACT START */
316 static void timodrsrv(queue_t *);
324 /* ONC_PLUS EXTRACT END */
317 static void timodwsrv(queue_t *);
326 /* ONC_PLUS EXTRACT START */
318 static int timodrproc(queue_t *, mblk_t *);
319 static int timodwproc(queue_t *, mblk_t *);
329 /* ONC_PLUS EXTRACT END */

321 /* stream data structure definitions */

323 static struct module_info timod_info =
324         {TIMOD_ID, "timod", 0, INFPSZ, 512, 128};
325 static struct qinit timodrinit = {
326         (int (*)())timodrput,
327         (int (*)())timodrsrv,
328         timodopen,
329         timodclose,
330         nulldev,
331         &timod_info,
332         NULL
333 };
```

*unchanged_portion_omitted_*

```
343 static struct streamtab timinfo = { &timodrinit, &timodwinit, NULL, NULL };

355 /* ONC_PLUS EXTRACT START */
345 /*
346  * timodopen -  open routine gets called when the module gets pushed
347  *              onto the stream.
348  */
349 /*ARGSUSED*/
350 static int
351 timodopen(
352         queue_t *q,
353         dev_t *devp,
354         int flag,
355         int sflag,
356         cred_t *crp)
357 {
358         struct tim_tim *tp;
359         struct stroptions *sop;
360         mblk_t *bp;

362         ASSERT(q != NULL);

364         if (q->q_ptr) {
365                 return (0);
366         }

368         if ((bp = allocb(sizeof (struct stroptions), BPRI_MED)) == 0)
369                 return (ENOMEM);

371         tp = kmem_zalloc(sizeof (struct tim_tim), KM_SLEEP);

373         tp->tim_cpid = -1;
```

```
374         tp->tim_saved_prim = -1;

376         mutex_init(&tp->tim_mutex, NULL, MUTEX_DEFAULT, NULL);

378         q->q_ptr = (caddr_t)tp;
379         WR(q)->q_ptr = (caddr_t)tp;

381         tilogp("timodopen: Allocated for tp %lx\n", (uintptr_t)tp);
382         tilogp("timodopen: Allocated for q %lx\n", (uintptr_t)q);

384         /* Must be done before tpi_findprov and _ILP32 q_next walk below */
385         qprocson(q);

387         tp->tim_provinfo = tpi_findprov(q);

389         /*
390          * Defer allocation of the buffers for the local address and
391          * the peer's address until we need them.
392          * Assume that timod has to handle getname until we here
393          * an iocack from the transport provider or we know that
394          * transport provider doesn't understand it.
395          */
396         if (tp->tim_provinfo->tpi_myname != PI_YES) {
397                 TILOG("timodopen: setting DO_MYNAME\n", 0);
398                 tp->tim_flags |= DO_MYNAME;
399         }

401         if (tp->tim_provinfo->tpi_peername != PI_YES) {
402                 TILOG("timodopen: setting DO_PEERNAME\n", 0);
403                 tp->tim_flags |= DO_PEERNAME;
404         }

406 #ifdef  _ILP32
407         {
408                 queue_t *driverq;

410                 /*
411                  * Find my driver's read queue (for T_CONN_RES handling)
412                  */
413                 driverq = WR(q);
414                 while (SAMESTR(driverq))
415                         driverq = driverq->q_next;

417                 tp->tim_acceptor = (t_uscalar_t)RD(driverq);
418         }
419 #else
420         tp->tim_acceptor = (t_uscalar_t)getminor(*devp);
421 #endif  /* _ILP32 */

423         /*
424          * Add this one to the list.
425          */
426         tim_addlink(tp);

428         /*
429          * Send M_SETOPTS to stream head to make sure M_PCPROTO messages
430          * are not flushed. This prevents application deadlocks.
431          */
432         bp->b_datap->db_type = M_SETOPTS;
433         bp->b_wptr += sizeof (struct stroptions);
434         sop = (struct stroptions *)bp->b_rptr;
435         sop->so_flags = SO_READOPT;
436         sop->so_readopt = RFLUSHPCPROT;

438         putnext(q, bp);
```



```
440         return (0);
441 }
_____unchanged_portion_omitted_
491 /* ONC_PLUS EXTRACT END */

481 /*
482  * timodclose - This routine gets called when the module gets popped
483  * off of the stream.
484  */
485 /*ARGSUSED*/
486 static int
487 timodclose(
488         queue_t *q,
489         int flag,
490         cred_t *crp)
491 {
492         struct tim_tim *tp;
493         mblk_t *mp;
494         mblk_t *nmp;

496         ASSERT(q != NULL);

498         tp = (struct tim_tim *)q->q_ptr;
499         q->q_ptr = NULL;

501         ASSERT(tp != NULL);

503         tilogp("timodclose: Entered for tp %lx\n", (uintptr_t)tp);
504         tilogp("timodclose: Entered for q %lx\n", (uintptr_t)q);

506         qprocsoff(q);
507         tim_dellink(tp);

509         /*
510          * Cancel any outstanding bufcall
511          * or timeout requests.
512          */
513         if (tp->tim_wbufcid) {
514                 qunbufcall(q, tp->tim_wbufcid);
515                 tp->tim_wbufcid = 0;
516         }
517         if (tp->tim_rbufcid) {
518                 qunbufcall(q, tp->tim_rbufcid);
519                 tp->tim_rbufcid = 0;
520         }
521         if (tp->tim_wtimoutid) {
522                 (void) quntimeout(q, tp->tim_wtimoutid);
523                 tp->tim_wtimoutid = 0;
524         }
525         if (tp->tim_rtimoutid) {
526                 (void) quntimeout(q, tp->tim_rtimoutid);
527                 tp->tim_rtimoutid = 0;
528         }

530         if (tp->tim_tcap_timoutid != 0) {
531                 (void) quntimeout(q, tp->tim_tcap_timoutid);
532                 tp->tim_tcap_timoutid = 0;
533         }

535         if (tp->tim_iocsave != NULL)
536                 freemsg(tp->tim_iocsave);
537         mp = tp->tim_consave;
538         while (mp) {
539                 nmp = mp->b_next;
540                 mp->b_next = NULL;
541                 freemsg(mp);
```

```
542                 mp = nmp;
543         }
544         ASSERT(tp->tim_mymaxlen >= 0);
545         if (tp->tim_mymaxlen != 0)
546                 kmem_free(tp->tim_myname, (size_t)tp->tim_mymaxlen);
547         ASSERT(tp->tim_peermaxlen >= 0);
548         if (tp->tim_peermaxlen != 0)
549                 kmem_free(tp->tim_peername, (size_t)tp->tim_peermaxlen);

551         q->q_ptr = WR(q)->q_ptr = NULL;

553         mutex_destroy(&tp->tim_mutex);

555         if (tp->tim_peercred != NULL)
556                 crfree(tp->tim_peercred);

558         kmem_free(tp, sizeof (struct tim_tim));

560         return (0);
561 }
```

unchanged_portion_omitted_

```
640 /* ONC_PLUS EXTRACT START */
628 /*
629  * timodrsrv -  Module read queue service procedure.  This is called when
630  *              messages are placed on an empty queue, when high priority
631  *              messages are placed on the queue, and when flow control
632  *              restrictions subside.  This code used to be included in a
633  *              put procedure, but it was moved to a service procedure
634  *              because several points were added where memory allocation
635  *              could fail, and there is no reasonable recovery mechanism
636  *              from the put procedure.
637  */
638 /*ARGSUSED*/
639 static void
640 timodrsrv(queue_t *q)
641 {
655 /* ONC_PLUS EXTRACT END */
642         mblk_t *mp;
643         struct tim_tim *tp;

645         ASSERT(q != NULL);

647         tp = (struct tim_tim *)q->q_ptr;
648         if (!tp)
649                 return;

651         while ((mp = getq(q)) != NULL) {
652                 if (timodrproc(q, mp)) {
653                         /*
654                          * timodrproc did a putbq - stop processing
655                          * messages.
656                          */
657                         return;
658                 }
659         }
674 /* ONC_PLUS EXTRACT START */
660 }
```

unchanged_portion_omitted_

```
680 static int
681 timodrproc(queue_t *q, mblk_t *mp)
682 {
683         uint32_t auditing = AU_AUDITING();
684         union T_primitives *pptr;
685         struct tim_tim *tp;
```



```
686         struct iocblk *iocbp;
687         mblk_t *nbp;
688         size_t blen;
704 /* ONC_PLUS EXTRACT END */

690         tp = (struct tim_tim *)q->q_ptr;

708 /* ONC_PLUS EXTRACT START */
692         switch (mp->b_datap->db_type) {
693         default:
694                 putnext(q, mp);
695                 break;

697         case M_ERROR:
698                 TILOG("timodrproc: Got M_ERROR, flags = %x\n", tp->tim_flags);
699                 /*
700                  * There is no specified standard response for driver when it
701                  * receives unknown message type and M_ERROR is one
702                  * possibility. If we send T_CAPABILITY_REQ down and transport
703                  * provider responds with M_ERROR we assume that it doesn't
704                  * understand this message type. This assumption may be
705                  * sometimes incorrect (transport may reply with M_ERROR for
706                  * some other reason) but there is no way for us to distinguish
707                  * between different cases. In the worst case timod and everyone
708                  * else sharing global transport description with it may end up
709                  * emulating T_CAPABILITY_REQ.
710                  */

712                 /*
713                  * Check that we are waiting for T_CAPABILITY_ACK and
714                  * T_CAPABILITY_REQ is not implemented by transport or emulated
715                  * by timod.
716                  */
717                 if ((tp->tim_provinfo->tpi_capability == PI_DONTKNOW) &&
718                     ((tp->tim_flags & TI_CAP_RECVD) != 0)) {
719                         /*
720                          * Good chances that this transport doesn't provide
721                          * T_CAPABILITY_REQ. Mark this information  permanently
722                          * for the module + transport combination.
723                          */
724                         PI_PROVLOCK(tp->tim_provinfo);
725                         if (tp->tim_provinfo->tpi_capability == PI_DONTKNOW)
726                                 tp->tim_provinfo->tpi_capability = PI_NO;
727                         PI_PROVUNLOCK(tp->tim_provinfo);
728                         if (tp->tim_tcap_timoutid != 0) {
729                                 (void) quntimeout(q, tp->tim_tcap_timoutid);
730                                 tp->tim_tcap_timoutid = 0;
731                         }
732                 }
733                 putnext(q, mp);
734                 break;
735         case M_DATA:
736                 if (!bcanputnext(q, mp->b_band)) {
737                         (void) putbq(q, mp);
738                         return (1);
739                 }
740                 putnext(q, mp);
741                 break;

743         case M_PROTO:
744         case M_PCPROTO:
745                 blen = MBLKL(mp);
746                 if (blen < sizeof (t_scalar_t)) {
747                         /*
748                          * Note: it's not actually possible to get
749                          * here with db_type M_PCPROTO, because
```

```
750                          * timodrput has already checked MBLKL, and
751                          * thus the assertion below.  If the length
752                          * was too short, then the message would have
753                          * already been putnext'd, and would thus
754                          * never appear here.  Just the same, the code
755                          * below handles the impossible case since
756                          * it's easy to do and saves future
757                          * maintainers from unfortunate accidents.
758                          */
759                         ASSERT(mp->b_datap->db_type == M_PROTO);
760                         if (mp->b_datap->db_type == M_PROTO &&
761                             !bcanputnext(q, mp->b_band)) {
762                                 (void) putbq(q, mp);
763                                 return (1);
764                         }
765                         putnext(q, mp);
766                         break;
767                 }

769                 pptr = (union T_primitives *)mp->b_rptr;
770                 switch (pptr->type) {
771                 default:
789 /* ONC_PLUS EXTRACT END */

773                         if (auditing)
774                                 audit_sock(T_UNITDATA_IND, q, mp, TIMOD_ID);
793 /* ONC_PLUS EXTRACT START */
775                         putnext(q, mp);
776                         break;
796 /* ONC_PLUS EXTRACT END */

778                 case T_ERROR_ACK:
779                         /* Restore db_type - recover() might have changed it */
780                         mp->b_datap->db_type = M_PCPROTO;
781                         if (blen < sizeof (struct T_error_ack)) {
782                                 putnext(q, mp);
783                                 break;
784                         }

786                         tilog("timodrproc: Got T_ERROR_ACK, flags = %x\n",
787                             tp->tim_flags);

789                         if ((tp->tim_flags & WAIT_CONNRESACK) &&
790                             tp->tim_saved_prim == pptr->error_ack.ERROR_prim) {
791                                 tp->tim_flags &=
792                                     ~(WAIT_CONNRESACK | WAITIOCACK);
793                                 freemsg(tp->tim_iocsave);
794                                 tp->tim_iocsave = NULL;
795                                 tp->tim_saved_prim = -1;
796                                 putnext(q, mp);
797                         } else if (tp->tim_flags & WAITIOCACK) {
798                                 tim_send_ioc_error_ack(q, tp, mp);
799                         } else {
800                                 putnext(q, mp);
801                         }
802                         break;

804                 case T_OK_ACK:
805                         if (blen < sizeof (pptr->ok_ack)) {
806                                 mp->b_datap->db_type = M_PCPROTO;
807                                 putnext(q, mp);
808                                 break;
809                         }

811                         tilog("timodrproc: Got T_OK_ACK\n", 0);
```



```
813                         if (pptr->ok_ack.CORRECT_prim == T_UNBIND_REQ)
814                                 tp->tim_mylen = 0;

816                         if ((tp->tim_flags & WAIT_CONNRESACK) &&
817                             tp->tim_saved_prim == pptr->ok_ack.CORRECT_prim) {
818                                 struct T_conn_res *resp;
819                                 struct T_conn_ind *indp;
820                                 struct tim_tim *ntp;
821                                 caddr_t ptr;

823                                 rw_enter(&tim_list_rwlock, RW_READER);
824                                 resp = (struct T_conn_res *)
825                                     tp->tim_iocsave->b_rptr;
826                                 ntp = tim_findlink(resp->ACCEPTOR_id);
827                                 if (ntp == NULL)
828                                         goto cresackout;

830                                 mutex_enter(&ntp->tim_mutex);
831                                 if (ntp->tim_peercred != NULL)
832                                         crfree(ntp->tim_peercred);
833                                 ntp->tim_peercred =
834                                     msg_getcred(tp->tim_iocsave->b_cont,
835                                     &ntp->tim_cpid);
836                                 if (ntp->tim_peercred != NULL)
837                                         crhold(ntp->tim_peercred);

839                                 if (!(ntp->tim_flags & DO_PEERNAME)) {
840                                         mutex_exit(&ntp->tim_mutex);
841                                         goto cresackout;
842                                 }

844                                 indp = (struct T_conn_ind *)
845                                     tp->tim_iocsave->b_cont->b_rptr;
846                                 /* true as message is put on list */
847                                 ASSERT(indp->SRC_length >= 0);

849                                 if (indp->SRC_length > ntp->tim_peermaxlen) {
850                                         ptr = kmem_alloc(indp->SRC_length,
851                                             KM_NOSLEEP);
852                                         if (ptr == NULL) {
853                                                 mutex_exit(&ntp->tim_mutex);
854                                                 rw_exit(&tim_list_rwlock);
855                                                 tilog("timodwproc: kmem_alloc "
856                                                     "failed, attempting "
857                                                     "recovery\n", 0);
858                                                 tim_recover(q, mp,
859                                                     indp->SRC_length);
860                                                 return (1);
861                                         }
862                                         if (ntp->tim_peermaxlen > 0)
863                                                 kmem_free(ntp->tim_peername,
864                                                     ntp->tim_peermaxlen);
865                                         ntp->tim_peername = ptr;
866                                         ntp->tim_peermaxlen = indp->SRC_length;
867                                 }
868                                 ntp->tim_peerlen = indp->SRC_length;
869                                 ptr = (caddr_t)indp + indp->SRC_offset;
870                                 bcopy(ptr, ntp->tim_peername, ntp->tim_peerlen);

872                                 mutex_exit(&ntp->tim_mutex);

874                         cresackout:
875                                 rw_exit(&tim_list_rwlock);
876                                 tp->tim_flags &=
877                                     ~(WAIT_CONNRESACK | WAITIOCACK);
878                                 freemsg(tp->tim_iocsave);
```

```
879                                 tp->tim_iocsave = NULL;
880                                 tp->tim_saved_prim = -1;
881                         }

883                         tim_send_reply(q, mp, tp, pptr->ok_ack.CORRECT_prim);
884                         break;

906 /* ONC_PLUS EXTRACT START */
886                 case T_BIND_ACK: {
887                         struct T_bind_ack *ackp =
888                             (struct T_bind_ack *)mp->b_rptr;

890                         /* Restore db_type - recover() might have changed it */
891                         mp->b_datap->db_type = M_PCPROTO;
892                         if (blen < sizeof (*ackp)) {
893                                 putnext(q, mp);
894                                 break;
895                         }

897                         /* save negotiated backlog */
898                         tp->tim_backlog = ackp->CONIND_number;

900                         if (((tp->tim_flags & WAITIOCACK) == 0) ||
901                             ((tp->tim_saved_prim != O_T_BIND_REQ) &&
902                             (tp->tim_saved_prim != T_BIND_REQ))) {
903                                 putnext(q, mp);
904                                 break;
905                         }
906                         ASSERT(tp->tim_iocsave != NULL);

908                         if (tp->tim_flags & DO_MYNAME) {
909                                 caddr_t p;

911                                 if (ackp->ADDR_length < 0 ||
912                                     mp->b_rptr + ackp->ADDR_offset +
913                                     ackp->ADDR_length > mp->b_wptr) {
914                                         putnext(q, mp);
915                                         break;
916                                 }
917                                 if (ackp->ADDR_length > tp->tim_mymaxlen) {
918                                         p = kmem_alloc(ackp->ADDR_length,
919                                             KM_NOSLEEP);
920                                         if (p == NULL) {
921                                                 tilog("timodrproc: kmem_alloc "
922                                                     "failed attempt recovery",
923                                                     0);

925                                                 tim_recover(q, mp,
926                                                     ackp->ADDR_length);
927                                                 return (1);
928                                         }
929                                         ASSERT(tp->tim_mymaxlen >= 0);
930                                         if (tp->tim_mymaxlen != NULL) {
931                                                 kmem_free(tp->tim_myname,
932                                                     tp->tim_mymaxlen);
933                                         }
934                                         tp->tim_myname = p;
935                                         tp->tim_mymaxlen = ackp->ADDR_length;
936                                 }
937                                 tp->tim_mylen = ackp->ADDR_length;
938                                 bcopy(mp->b_rptr + ackp->ADDR_offset,
939                                     tp->tim_myname, tp->tim_mylen);
940                         }
941                         tim_ioctl_send_reply(q, tp->tim_iocsave, mp);
942                         tp->tim_iocsave = NULL;
943                         tp->tim_saved_prim = -1;
```



```
944                         tp->tim_flags &= ~(WAITIOCACK | WAIT_IOCINFOACK |
945                             TI_CAP_RECVD | CAP_WANTS_INFO);
946                         break;
947                 }

970 /* ONC_PLUS EXTRACT END */
949                 case T_OPTMGMT_ACK:

951                         tilog("timodrproc: Got T_OPTMGMT_ACK\n", 0);

953                         /* Restore db_type - recover() might have change it */
954                         mp->b_datap->db_type = M_PCPROTO;

956                         if (((tp->tim_flags & WAITIOCACK) == 0) ||
957                             ((tp->tim_saved_prim != T_SVR4_OPTMGMT_REQ) &&
958                             (tp->tim_saved_prim != T_OPTMGMT_REQ))) {
959                                 putnext(q, mp);
960                         } else {
961                                 ASSERT(tp->tim_iocsave != NULL);
962                                 tim_ioctl_send_reply(q, tp->tim_iocsave, mp);
963                                 tp->tim_iocsave = NULL;
964                                 tp->tim_saved_prim = -1;
965                                 tp->tim_flags &= ~(WAITIOCACK |
966                                     WAIT_IOCINFOACK | TI_CAP_RECVD |
967                                     CAP_WANTS_INFO);
968                         }
969                 break;

971                 case T_INFO_ACK: {
972                 struct T_info_ack *tia = (struct T_info_ack *)pptr;

974                 /* Restore db_type - recover() might have changed it */
975                 mp->b_datap->db_type = M_PCPROTO;

977                 if (blen < sizeof (*tia)) {
978                         putnext(q, mp);
979                         break;
980                 }

982                 tilog("timodrproc: Got T_INFO_ACK, flags = %x\n",
983                     tp->tim_flags);

985                 timodprocessinfo(q, tp, tia);

987                 TILOG("timodrproc: flags = %x\n", tp->tim_flags);
988                 if ((tp->tim_flags & WAITIOCACK) != 0) {
989                         size_t  expected_ack_size;
990                         ssize_t deficit;
991                         int     ioc_cmd;
992                         struct T_capability_ack *tcap;

994                         /*
995                          * The only case when T_INFO_ACK may be received back
996                          * when we are waiting for ioctl to complete is when
997                          * this ioctl sent T_INFO_REQ down.
998                          */
999                         if (!(tp->tim_flags & WAIT_IOCINFOACK)) {
1000                                putnext(q, mp);
1001                                break;
1002                        }
1003                        ASSERT(tp->tim_iocsave != NULL);

1005                        iocbp = (struct iocblk *)tp->tim_iocsave->b_rptr;
1006                        ioc_cmd = iocbp->ioc_cmd;

1008                        /*
```

```
1009                         * Was it sent from TI_CAPABILITY emulation?
1010                         */
1011                        if (ioc_cmd == TI_CAPABILITY) {
1012                                struct T_info_ack       saved_info;

1014                                /*
1015                                 * Perform sanity checks. The only case when we
1016                                 * send T_INFO_REQ from TI_CAPABILITY is when
1017                                 * timod emulates T_CAPABILITY_REQ and CAP_bits1
1018                                 * has TC1_INFO set.
1019                                 */
1020                                if ((tp->tim_flags &
1021                                    (TI_CAP_RECVD | CAP_WANTS_INFO)) !=
1022                                    (TI_CAP_RECVD | CAP_WANTS_INFO)) {
1023                                        putnext(q, mp);
1024                                        break;
1025                                }

1027                                TILOG("timodrproc: emulating TI_CAPABILITY/"
1028                                    "info\n", 0);

1030                                /* Save info & reuse mp for T_CAPABILITY_ACK */
1031                                saved_info = *tia;

1033                                mp = tpi_ack_alloc(mp,
1034                                    sizeof (struct T_capability_ack),
1035                                    M_PCPROTO, T_CAPABILITY_ACK);

1037                                if (mp == NULL) {
1038                                        tilog("timodrproc: realloc failed, "
1039                                            "no recovery attempted\n", 0);
1040                                        return (1);
1041                                }

1043                                /*
1044                                 * Copy T_INFO information into T_CAPABILITY_ACK
1045                                 */
1046                                tcap = (struct T_capability_ack *)mp->b_rptr;
1047                                tcap->CAP_bits1 = TC1_INFO;
1048                                tcap->INFO_ack = saved_info;
1049                                tp->tim_flags &= ~(WAITIOCACK |
1050                                    WAIT_IOCINFOACK | TI_CAP_RECVD |
1051                                    CAP_WANTS_INFO);
1052                                tim_ioctl_send_reply(q, tp->tim_iocsave, mp);
1053                                tp->tim_iocsave = NULL;
1054                                tp->tim_saved_prim = -1;
1055                                break;
1056                        }

1058                        /*
1059                         * The code for TI_SYNC/TI_GETINFO is left here only for
1060                         * backward compatibility with staticaly linked old
1061                         * applications. New TLI/XTI code should use
1062                         * TI_CAPABILITY for getting transport info and should
1063                         * not use TI_GETINFO/TI_SYNC for this purpose.
1064                         */

1066                        /*
1067                         * make sure the message sent back is the size of
1068                         * the "expected ack"
1069                         * For TI_GETINFO, expected ack size is
1070                         *      sizeof (T_info_ack)
1071                         * For TI_SYNC, expected ack size is
1072                         *      sizeof (struct ti_sync_ack);
1073                         */
1074                        if (ioc_cmd != TI_GETINFO && ioc_cmd != TI_SYNC) {
```



```
1075                                putnext(q, mp);
1076                                break;
1077                        }

1079                        expected_ack_size =
1080                            sizeof (struct T_info_ack); /* TI_GETINFO */
1081                        if (iocbp->ioc_cmd == TI_SYNC) {
1082                                expected_ack_size = 2 * sizeof (uint32_t) +
1083                                    sizeof (struct ti_sync_ack);
1084                        }
1085                        deficit = expected_ack_size - blen;

1087                        if (deficit != 0) {
1088                                if (mp->b_datap->db_lim - mp->b_wptr <
1089                                    deficit) {
1090                                        mblk_t *tmp = allocb(expected_ack_size,
1091                                            BPRI_HI);
1092                                        if (tmp == NULL) {
1093                                                ASSERT(MBLKSIZE(mp) >=
1094                                                    sizeof (struct T_error_ack))

1096                                                tilog("timodrproc: allocb failed
1097                                                    "recovery attempt\n", 0);

1099                                                mp->b_rptr = mp->b_datap->db_bas
1100                                                pptr = (union T_primitives *)
1101                                                    mp->b_rptr;
1102                                                pptr->error_ack.ERROR_prim = T_I
1103                                                pptr->error_ack.TLI_error = TSYS
1104                                                pptr->error_ack.UNIX_error = EAG
1105                                                pptr->error_ack.PRIM_type = T_ER
1106                                                mp->b_datap->db_type = M_PCPROTO
1107                                                tim_send_ioc_error_ack(q, tp, mp
1108                                                break;
1109                                        } else {
1110                                                bcopy(mp->b_rptr, tmp->b_rptr, b
1111                                                tmp->b_wptr += blen;
1112                                                pptr = (union T_primitives *)
1113                                                    tmp->b_rptr;
1114                                                freemsg(mp);
1115                                                mp = tmp;
1116                                        }
1117                                }
1118                        }
1119                        /*
1120                         * We now have "mp" which has enough space for an
1121                         * appropriate ack and contains struct T_info_ack
1122                         * that the transport provider returned. We now
1123                         * stuff it with more stuff to fullfill
1124                         * TI_SYNC ioctl needs, as necessary
1125                         */
1126                        if (iocbp->ioc_cmd == TI_SYNC) {
1127                                /*
1128                                 * Assumes struct T_info_ack is first embedded
1129                                 * type in struct ti_sync_ack so it is
1130                                 * automatically there.
1131                                 */
1132                                struct ti_sync_ack *tsap =
1133                                    (struct ti_sync_ack *)mp->b_rptr;

1135                                /*
1136                                 * tsap->tsa_qlen needs to be set only if
1137                                 * TSRF_QLEN_REQ flag is set, but for
1138                                 * compatibility with statically linked
1139                                 * applications it is set here regardless of the
1140                                 * flag since old XTI library expected it to be
```

```
1141                                 * set.
1142                                 */
1143                                tsap->tsa_qlen = tp->tim_backlog;
1144                                tsap->tsa_flags = 0x0; /* intialize clear */
1145                                if (tp->tim_flags & PEEK_RDQ_EXPIND) {
1146                                        /*
1147                                         * Request to peek for EXPIND in
1148                                         * rcvbuf.
1149                                         */
1150                                        if (ti_expind_on_rdqueues(q)) {
1151                                                /*
1152                                                 * Expedited data is
1153                                                 * queued on the stream
1154                                                 * read side
1155                                                 */
1156                                                tsap->tsa_flags |=
1157                                                    TSAF_EXP_QUEUED;
1158                                        }
1159                                        tp->tim_flags &=
1160                                            ~PEEK_RDQ_EXPIND;
1161                                }
1162                                mp->b_wptr += 2*sizeof (uint32_t);
1163                        }
1164                        tim_ioctl_send_reply(q, tp->tim_iocsave, mp);
1165                        tp->tim_iocsave = NULL;
1166                        tp->tim_saved_prim = -1;
1167                        tp->tim_flags &= ~(WAITIOCACK | WAIT_IOCINFOACK |
1168                            TI_CAP_RECVD | CAP_WANTS_INFO);
1169                        break;
1170                }
1171            }

1173            putnext(q, mp);
1174            break;

1176            case T_ADDR_ACK:
1177                tilog("timodrproc: Got T_ADDR_ACK\n", 0);
1178                tim_send_reply(q, mp, tp, T_ADDR_REQ);
1179                break;

1203 /* ONC_PLUS EXTRACT START */
1181                case T_CONN_IND: {
1182                        struct T_conn_ind *tcip =
1183                            (struct T_conn_ind *)mp->b_rptr;

1185                        tilog("timodrproc: Got T_CONN_IND\n", 0);

1187                        if (blen >= sizeof (*tcip) &&
1188                            MBLKIN(mp, tcip->SRC_offset, tcip->SRC_length)) {
1189                                if (((nbp = dupmsg(mp)) != NULL) ||
1190                                    ((nbp = copymsg(mp)) != NULL)) {
1191                                        nbp->b_next = tp->tim_consave;
1192                                        tp->tim_consave = nbp;
1193                                } else {
1194                                        tim_recover(q, mp,
1195                                            (t_scalar_t)sizeof (mblk_t));
1196                                        return (1);
1197                                }
1198                        }
1222 /* ONC_PLUS EXTRACT END */
1199                        if (auditing)
1200                                audit_sock(T_CONN_IND, q, mp, TIMOD_ID);
1225 /* ONC_PLUS EXTRACT START */
1201                        putnext(q, mp);
1202                        break;
1203                }
```



```
1230 /* ONC_PLUS EXTRACT END */
1205            case T_CONN_CON:
1206                mutex_enter(&tp->tim_mutex);
1207                if (tp->tim_peercred != NULL)
1208                        crfree(tp->tim_peercred);
1209                tp->tim_peercred = msg_getcred(mp, &tp->tim_cpid);
1210                if (tp->tim_peercred != NULL)
1211                        crhold(tp->tim_peercred);
1212                mutex_exit(&tp->tim_mutex);

1214                tilog("timodrproc: Got T_CONN_CON\n", 0);

1216                tp->tim_flags &= ~CONNWAIT;
1217                putnext(q, mp);
1218                break;

1220            case T_DISCON_IND: {
1221                struct T_discon_ind *disp;
1222                struct T_conn_ind *conp;
1223                mblk_t *pbp = NULL;

1225                if (q->q_first != 0)
1226                        tilog("timodrput: T_DISCON_IND - flow control\n", 0);

1228                if (blen < sizeof (*disp)) {
1229                        putnext(q, mp);
1230                        break;
1231                }

1233                disp = (struct T_discon_ind *)mp->b_rptr;

1235                tilog("timodrproc: Got T_DISCON_IND Reason: %d\n",
1236                    disp->DISCON_reason);

1238                tp->tim_flags &= ~(CONNWAIT|LOCORDREL|REMORDREL);
1239                tim_clear_peer(tp);
1240                for (nbp = tp->tim_consave; nbp; nbp = nbp->b_next) {
1241                        conp = (struct T_conn_ind *)nbp->b_rptr;
1242                        if (conp->SEQ_number == disp->SEQ_number)
1243                                break;
1244                        pbp = nbp;
1245                }
1246                if (nbp) {
1247                        if (pbp)
1248                                pbp->b_next = nbp->b_next;
1249                        else
1250                                tp->tim_consave = nbp->b_next;
1251                        nbp->b_next = NULL;
1252                        freemsg(nbp);
1253                }
1254                putnext(q, mp);
1255                break;
1256            }

1258            case T_ORDREL_IND:

1260                    tilog("timodrproc: Got T_ORDREL_IND\n", 0);

1262                    if (tp->tim_flags & LOCORDREL) {
1263                            tp->tim_flags &= ~(LOCORDREL|REMORDREL);
1264                            tim_clear_peer(tp);
1265                    } else {
1266                            tp->tim_flags |= REMORDREL;
1267                    }
1268                    putnext(q, mp);
```

```
1269                    break;

1271            case T_EXDATA_IND:
1272            case T_DATA_IND:
1273            case T_UNITDATA_IND:
1274                if (pptr->type == T_EXDATA_IND)
1275                        tilog("timodrproc: Got T_EXDATA_IND\n", 0);

1277                if (!bcanputnext(q, mp->b_band)) {
1278                        (void) putbq(q, mp);
1279                        return (1);
1280                }
1281                putnext(q, mp);
1282                break;

1284            case T_CAPABILITY_ACK: {
1285                    struct T_capability_ack *tca;

1287                    if (blen < sizeof (*tca)) {
1288                            putnext(q, mp);
1289                            break;
1290                    }

1292                    /* This transport supports T_CAPABILITY_REQ */
1293                    tilog("timodrproc: Got T_CAPABILITY_ACK\n", 0);

1295                    PI_PROVLOCK(tp->tim_provinfo);
1296                    if (tp->tim_provinfo->tpi_capability != PI_YES)
1297                            tp->tim_provinfo->tpi_capability = PI_YES;
1298                    PI_PROVUNLOCK(tp->tim_provinfo);

1300                    /* Reset possible pending timeout */
1301                    if (tp->tim_tcap_timoutid != 0) {
1302                            (void) quntimeout(q, tp->tim_tcap_timoutid);
1303                            tp->tim_tcap_timoutid = 0;
1304                    }

1306                    tca = (struct T_capability_ack *)mp->b_rptr;

1308                    if (tca->CAP_bits1 & TC1_INFO)
1309                            timodprocessinfo(q, tp, &tca->INFO_ack);

1311                    tim_send_reply(q, mp, tp, T_CAPABILITY_REQ);
1312                }
1313                break;
1314            }
1315            break;

1343 /* ONC_PLUS EXTRACT START */
1317        case M_FLUSH:

1319                tilog("timodrproc: Got M_FLUSH\n", 0);

1321                if (*mp->b_rptr & FLUSHR) {
1322                        if (*mp->b_rptr & FLUSHBAND)
1323                                flushband(q, *(mp->b_rptr + 1), FLUSHDATA);
1324                        else
1325                                flushq(q, FLUSHDATA);
1326                }
1327                putnext(q, mp);
1328                break;
1356 /* ONC_PLUS EXTRACT END */

1330        case M_IOCACK:
1331            iocbp = (struct iocblk *)mp->b_rptr;
```



```
1333            tilog("timodrproc: Got M_IOCACK\n", 0);

1335            if (iocbp->ioc_cmd == TI_GETMYNAME) {

1337                /*
1338                 * Transport provider supports this ioctl,
1339                 * so I don't have to.
1340                 */
1341                if ((tp->tim_flags & DO_MYNAME) != 0) {
1342                        tp->tim_flags &= ~DO_MYNAME;
1343                        PI_PROVLOCK(tp->tim_provinfo);
1344                        tp->tim_provinfo->tpi_myname = PI_YES;
1345                        PI_PROVUNLOCK(tp->tim_provinfo);
1346                }

1348                ASSERT(tp->tim_mymaxlen >= 0);
1349                if (tp->tim_mymaxlen != 0) {
1350                        kmem_free(tp->tim_myname, (size_t)tp->tim_mymaxlen);
1351                        tp->tim_myname = NULL;
1352                        tp->tim_mymaxlen = 0;
1353                }
1354                /* tim_iocsave may already be overwritten. */
1355                if (tp->tim_saved_prim == -1) {
1356                        freemsg(tp->tim_iocsave);
1357                        tp->tim_iocsave = NULL;
1358                }
1359            } else if (iocbp->ioc_cmd == TI_GETPEERNAME) {
1360                boolean_t clearit;

1362                /*
1363                 * Transport provider supports this ioctl,
1364                 * so I don't have to.
1365                 */
1366                if ((tp->tim_flags & DO_PEERNAME) != 0) {
1367                        tp->tim_flags &= ~DO_PEERNAME;
1368                        PI_PROVLOCK(tp->tim_provinfo);
1369                        tp->tim_provinfo->tpi_peername = PI_YES;
1370                        PI_PROVUNLOCK(tp->tim_provinfo);
1371                }

1373                mutex_enter(&tp->tim_mutex);
1374                ASSERT(tp->tim_peermaxlen >= 0);
1375                clearit = tp->tim_peermaxlen != 0;
1376                if (clearit) {
1377                        kmem_free(tp->tim_peername, tp->tim_peermaxlen);
1378                        tp->tim_peername = NULL;
1379                        tp->tim_peermaxlen = 0;
1380                        tp->tim_peerlen = 0;
1381                }
1382                mutex_exit(&tp->tim_mutex);
1383                if (clearit) {
1384                        mblk_t *bp;

1386                        bp = tp->tim_consave;
1387                        while (bp != NULL) {
1388                                nbp = bp->b_next;
1389                                bp->b_next = NULL;
1390                                freemsg(bp);
1391                                bp = nbp;
1392                        }
1393                        tp->tim_consave = NULL;
1394                }
1395                /* tim_iocsave may already be overwritten. */
1396                if (tp->tim_saved_prim == -1) {
1397                        freemsg(tp->tim_iocsave);
1398                        tp->tim_iocsave = NULL;
```

```
1399                }
1400            }
1401            putnext(q, mp);
1402            break;

1432 /* ONC_PLUS EXTRACT START */
1404        case M_IOCNAK:

1406                tilog("timodrproc: Got M_IOCNAK\n", 0);

1408                iocbp = (struct iocblk *)mp->b_rptr;
1409                if (((iocbp->ioc_cmd == TI_GETMYNAME) ||
1410                    (iocbp->ioc_cmd == TI_GETPEERNAME)) &&
1411                    ((iocbp->ioc_error == EINVAL) || (iocbp->ioc_error == 0))) {
1412                        PI_PROVLOCK(tp->tim_provinfo);
1413                        if (iocbp->ioc_cmd == TI_GETMYNAME) {
1414                                if (tp->tim_provinfo->tpi_myname == PI_DONTKNOW)
1415                                        tp->tim_provinfo->tpi_myname = PI_NO;
1416                        } else if (iocbp->ioc_cmd == TI_GETPEERNAME) {
1417                                if (tp->tim_provinfo->tpi_peername == PI_DONTKNO
1418                                        tp->tim_provinfo->tpi_peername = PI_NO;
1419                        }
1420                        PI_PROVUNLOCK(tp->tim_provinfo);
1421                        /* tim_iocsave may already be overwritten. */
1422                        if ((tp->tim_iocsave != NULL) &&
1423                            (tp->tim_saved_prim == -1)) {
1424                                freemsg(mp);
1425                                mp = tp->tim_iocsave;
1426                                tp->tim_iocsave = NULL;
1427                                tp->tim_flags |= NAMEPROC;
1428                                if (ti_doname(WR(q), mp) != DONAME_CONT) {
1429                                        tp->tim_flags &= ~NAMEPROC;
1430                                }
1431                                break;
1432                        }
1433                }
1434                putnext(q, mp);
1435                break;
1465 /* ONC_PLUS EXTRACT END */
1436        }

1438        return (0);
1439 }

1471 /* ONC_PLUS EXTRACT START */
1441 /*
1442  * timodwput -  Module write put procedure.  This is called from
1443  *              the module, driver, or stream head upstream/downstream.
1444  *              Handles M_FLUSH, M_DATA and some M_PROTO (T_DATA_REQ,
1445  *              and T_UNITDATA_REQ) messages. All others are queued to
1446  *              be handled by the service procedures.
1447  */

1449 static void
1450 timodwput(queue_t *q, mblk_t *mp)
1451 {
1452        union T_primitives *pptr;
1453        struct tim_tim *tp;
1454        struct iocblk *iocbp;

1456        /*
1457         * Enqueue normal-priority messages if our queue already
1458         * holds some messages for deferred processing but don't
1459         * enqueue those M_IOCTLs which will result in an
1460         * M_PCPROTO (ie, high priority) message being created.
1461         */
```



```
1493 /* ONC_PLUS EXTRACT END */
1462        if (q->q_first != 0 && mp->b_datap->db_type < QPCTL) {
1463                if (mp->b_datap->db_type == M_IOCTL) {
1464                        iocbp = (struct iocblk *)mp->b_rptr;
1465                        switch (iocbp->ioc_cmd) {
1466                        default:
1467                                (void) putq(q, mp);
1468                                return;

1470                        case TI_GETINFO:
1471                        case TI_SYNC:
1472                        case TI_CAPABILITY:
1473                                break;
1474                        }
1475                } else {
1476                        (void) putq(q, mp);
1477                        return;
1478                }
1479        }
1512 /* ONC_PLUS EXTRACT START */
1480        /*
1481         * Inline processing of data (to avoid additional procedure call).
1482         * Rest is handled in timodwproc.
1483         */

1485        switch (mp->b_datap->db_type) {
1486        case M_DATA:
1487                tp = (struct tim_tim *)q->q_ptr;
1488                ASSERT(tp);
1489                if (tp->tim_flags & CLTS) {
1490                        mblk_t  *tmp;

1492                        if ((tmp = tim_filladdr(q, mp, B_FALSE)) == NULL) {
1493                                (void) putq(q, mp);
1494                                break;
1495                        } else {
1496                                mp = tmp;
1497                        }
1498                }
1499                if (bcanputnext(q, mp->b_band))
1500                        putnext(q, mp);
1501                else
1502                        (void) putq(q, mp);
1503                break;
1504        case M_PROTO:
1505        case M_PCPROTO:
1506                pptr = (union T_primitives *)mp->b_rptr;
1507                switch (pptr->type) {
1541 /* ONC_PLUS EXTRACT END */
1508                case T_UNITDATA_REQ:
1509                        tp = (struct tim_tim *)q->q_ptr;
1510                        ASSERT(tp);
1511                        if (tp->tim_flags & CLTS) {
1512                                mblk_t  *tmp;

1514                                tmp = tim_filladdr(q, mp, B_FALSE);
1515                                if (tmp == NULL) {
1516                                        (void) putq(q, mp);
1517                                        break;
1518                                } else {
1519                                        mp = tmp;
1520                                }
1521                        }
1522                        if (bcanputnext(q, mp->b_band))
1523                                putnext(q, mp);
1524                        else
```

```
1525                                (void) putq(q, mp);
1526                        break;

1528                case T_DATA_REQ:
1529                case T_EXDATA_REQ:
1530                        if (bcanputnext(q, mp->b_band))
1531                                putnext(q, mp);
1532                        else
1533                                (void) putq(q, mp);
1534                        break;
1535                default:
1536                        (void) timodwproc(q, mp);
1537                        break;
1538                }
1539                break;
1574 /* ONC_PLUS EXTRACT START */
1540        default:
1541                (void) timodwproc(q, mp);
1542                break;
1543        }
1544 }
_____unchanged_portion_omitted_

1575 /*
1576  * Common routine to process write side messages
1577  */

1579 static int
1580 timodwproc(queue_t *q, mblk_t *mp)
1581 {
1582        union T_primitives *pptr;
1583        struct tim_tim *tp;
1584        uint32_t auditing = AU_AUDITING();
1585        mblk_t *tmp;
1586        struct iocblk *iocbp;
1587        int error;

1589        tp = (struct tim_tim *)q->q_ptr;

1591        switch (mp->b_datap->db_type) {
1592        default:
1593                putnext(q, mp);
1594                break;
1630 /* ONC_PLUS EXTRACT END */

1596        case M_DATA:
1597                if (tp->tim_flags & CLTS) {
1598                        if ((tmp = tim_filladdr(q, mp, B_TRUE)) == NULL) {
1599                                return (1);
1600                        } else {
1601                                mp = tmp;
1602                        }
1603                }
1604                if (!bcanputnext(q, mp->b_band)) {
1605                        (void) putbq(q, mp);
1606                        return (1);
1607                }
1608                putnext(q, mp);
1609                break;

1647 /* ONC_PLUS EXTRACT START */
1611        case M_IOCTL:

1613                iocbp = (struct iocblk *)mp->b_rptr;
1614                TILOG("timodwproc: Got M_IOCTL(%d)\n", iocbp->ioc_cmd);
```



```
1616                ASSERT(MBLKL(mp) == sizeof (struct iocblk));

1618                /*
1619                 * TPI requires we await response to a previously sent message
1620                 * before handling another, put it back on the head of queue.
1621                 * Since putbq() may see QWANTR unset when called from the
1622                 * service procedure, the queue must be explicitly scheduled
1623                 * for service, as no backenable will occur for this case.
1624                 * tim_ioctl_retry() sets a timer to handle the qenable.
1625                 */
1626                if (tp->tim_flags & WAITIOCACK) {
1627                        TILOG("timodwproc: putbq M_IOCTL(%d)\n",
1628                            iocbp->ioc_cmd);
1629                        (void) putbq(q, mp);
1630                        /* Called from timodwsrv() and messages on queue */
1631                        if (!(q->q_flag & QWANTR))
1632                                tim_ioctl_retry(q);
1633                        return (1);
1634                }
1672 /* ONC_PLUS EXTRACT END */

1636                switch (iocbp->ioc_cmd) {
1637                default:
1638                        putnext(q, mp);
1639                        break;

1641                case _I_GETPEERCRED:
1642                        if ((tp->tim_flags & COTS) == 0) {
1643                                miocnak(q, mp, 0, ENOTSUP);
1644                        } else {
1645                                mblk_t *cmp = mp->b_cont;
1646                                k_peercred_t *kp = NULL;

1648                                mutex_enter(&tp->tim_mutex);
1649                                if (cmp != NULL &&
1650                                    iocbp->ioc_flag == IOC_NATIVE &&
1651                                    (tp->tim_flags &
1652                                    (CONNWAIT|LOCORDREL|REMORDREL)) == 0 &&
1653                                    tp->tim_peercred != NULL &&
1654                                    DB_TYPE(cmp) == M_DATA &&
1655                                    MBLKL(cmp) == sizeof (k_peercred_t)) {
1656                                        kp = (k_peercred_t *)cmp->b_rptr;
1657                                        crhold(kp->pc_cr = tp->tim_peercred);
1658                                        kp->pc_cpid = tp->tim_cpid;
1659                                }
1660                                mutex_exit(&tp->tim_mutex);
1661                                if (kp != NULL)
1662                                        miocack(q, mp, sizeof (*kp), 0);
1663                                else
1664                                        miocnak(q, mp, 0, ENOTCONN);
1665                        }
1666                        break;
1667                case TI_BIND:
1668                case TI_UNBIND:
1669                case TI_OPTMGMT:
1670                case TI_GETADDRS:
1671                        TILOG("timodwproc: TI_{BIND|UNBIND|OPTMGMT|GETADDRS}"
1672                            "\n", 0);

1674                        /*
1675                         * We know that tim_send_ioctl_tpi_msg() is only
1676                         * going to examine the 'type' field, so we only
1677                         * check that we can access that much data.
1678                         */
1679                        error = miocpullup(mp, sizeof (t_scalar_t));
1680                        if (error != 0) {
```

```
1681                                    miocnak(q, mp, 0, error);
1682                                    break;
1683                            }
1684                            tim_send_ioctl_tpi_msg(q, mp, tp, iocbp);
1685                            break;

1687                    case TI_GETINFO:
1688                            TILOG("timodwproc: TI_GETINFO\n", 0);
1689                            error = miocpullup(mp, sizeof (struct T_info_req));
1690                            if (error != 0) {
1691                                    miocnak(q, mp, 0, error);
1692                                    break;
1693                            }
1694                            tp->tim_flags |= WAIT_IOCINFOACK;
1695                            tim_send_ioctl_tpi_msg(q, mp, tp, iocbp);
1696                            break;

1698                    case TI_SYNC: {
1699                            mblk_t *tsr_mp;
1700                            struct ti_sync_req *tsr;
1701                            uint32_t tsr_flags;

1703                            error = miocpullup(mp, sizeof (struct ti_sync_req));
1704                            if (error != 0) {
1705                                    miocnak(q, mp, 0, error);
1706                                    break;
1707                            }

1709                            tsr_mp = mp->b_cont;
1710                            tsr = (struct ti_sync_req *)tsr_mp->b_rptr;
1711                            TILOG("timodwproc: TI_SYNC(%x)\n", tsr->tsr_flags);

1713                            /*
1714                             * Save out the value of tsr_flags, in case we
1715                             * reallocb() tsr_mp (below).
1716                             */
1717                            tsr_flags = tsr->tsr_flags;
1718                            if ((tsr_flags & TSRF_INFO_REQ) == 0) {
1719                                    mblk_t *ack_mp = reallocb(tsr_mp,
1720                                        sizeof (struct ti_sync_ack), 0);

1722                                    /* Can reply immediately. */
1723                                    mp->b_cont = NULL;
1724                                    if (ack_mp == NULL) {
1725                                            tilog("timodwproc: allocb failed no "
1726                                                "recovery attempt\n", 0);
1727                                            freemsg(tsr_mp);
1728                                            miocnak(q, mp, 0, ENOMEM);
1729                                    } else {
1730                                            tim_answer_ti_sync(q, mp, tp,
1731                                                ack_mp, tsr_flags);
1732                                    }
1733                                    break;
1734                            }

1736                            /*
1737                             * This code is retained for compatibility with
1738                             * old statically linked applications. New code
1739                             * should use TI_CAPABILITY for all TPI
1740                             * information and should not use TSRF_INFO_REQ
1741                             * flag.
1742                             *
1743                             * defer processsing necessary to rput procedure
1744                             * as we need to get information from transport
1745                             * driver. Set flags that will tell the read
1746                             * side the work needed on this request.
```



```
1747                             */

1749                            if (tsr_flags & TSRF_IS_EXP_IN_RCVBUF)
1750                                    tp->tim_flags |= PEEK_RDQ_EXPIND;

1752                            /*
1753                             * Convert message to a T_INFO_REQ message; relies
1754                             * on sizeof (struct ti_sync_req) >= sizeof (struct
1755                             * T_info_req)).
1756                             */
1757                            ASSERT(MBLKL(tsr_mp) >= sizeof (struct T_info_req));

1759                            ((struct T_info_req *)tsr_mp->b_rptr)->PRIM_type =
1760                                T_INFO_REQ;
1761                            tsr_mp->b_wptr = tsr_mp->b_rptr +
1762                                sizeof (struct T_info_req);
1763                            tp->tim_flags |= WAIT_IOCINFOACK;
1764                            tim_send_ioctl_tpi_msg(q, mp, tp, iocbp);
1765                    }
1766                    break;

1768                    case TI_CAPABILITY: {
1769                            mblk_t *tcsr_mp;
1770                            struct T_capability_req *tcr;

1772                            error = miocpullup(mp, sizeof (*tcr));
1773                            if (error != 0) {
1774                                    miocnak(q, mp, 0, error);
1775                                    break;
1776                            }

1778                            tcsr_mp = mp->b_cont;
1779                            tcr = (struct T_capability_req *)tcsr_mp->b_rptr;
1780                            TILOG("timodwproc: TI_CAPABILITY(CAP_bits1 = %x)\n",
1781                                tcr->CAP_bits1);

1783                            if (tcr->PRIM_type != T_CAPABILITY_REQ) {
1784                                    TILOG("timodwproc: invalid msg type %d\n",
1785                                        tcr->PRIM_type);
1786                                    miocnak(q, mp, 0, EPROTO);
1787                                    break;
1788                            }

1790                            switch (tp->tim_provinfo->tpi_capability) {
1791                            case PI_YES:
1792                                    /* Just send T_CAPABILITY_REQ down */
1793                                    tim_send_ioctl_tpi_msg(q, mp, tp, iocbp);
1794                                    break;

1796                            case PI_DONTKNOW:
1797                                    /*
1798                                     * It is unknown yet whether transport provides
1799                                     * T_CAPABILITY_REQ or not. Send message down
1800                                     * and wait for reply.
1801                                     */

1803                                    ASSERT(tp->tim_tcap_timoutid == 0);
1804                                    if ((tcr->CAP_bits1 & TC1_INFO) == 0) {
1805                                            tp->tim_flags |= TI_CAP_RECVD;
1806                                    } else {
1807                                            tp->tim_flags |= (TI_CAP_RECVD |
1808                                                CAP_WANTS_INFO);
1809                                    }

1811                                    tp->tim_tcap_timoutid = qtimeout(q,
1812                                        tim_tcap_timer, q, tim_tcap_wait * hz);
```

```
1813                                    tim_send_ioctl_tpi_msg(q, mp, tp, iocbp);
1814                                    break;

1816                            case PI_NO:
1817                                    /*
1818                                     * Transport doesn't support T_CAPABILITY_REQ.
1819                                     * Either reply immediately or send T_INFO_REQ
1820                                     * if needed.
1821                                     */
1822                                    if ((tcr->CAP_bits1 & TC1_INFO) != 0) {
1823                                            tp->tim_flags |= (TI_CAP_RECVD |
1824                                                CAP_WANTS_INFO | WAIT_IOCINFOACK);
1825                                            TILOG("timodwproc: sending down "
1826                                                "T_INFO_REQ, flags = %x\n",
1827                                                tp->tim_flags);

1829                                    /*
1830                                     * Generate T_INFO_REQ message and send
1831                                     * it down
1832                                     */
1833                                            ((struct T_info_req *)tcsr_mp->b_rptr)->
1834                                                PRIM_type = T_INFO_REQ;
1835                                            tcsr_mp->b_wptr = tcsr_mp->b_rptr +
1836                                                sizeof (struct T_info_req);
1837                                            tim_send_ioctl_tpi_msg(q, mp, tp,
1838                                                iocbp);
1839                                            break;
1840                                    }



1843                                    /*
1844                                     * Can reply immediately. Just send back
1845                                     * T_CAPABILITY_ACK with CAP_bits1 set to 0.
1846                                     */
1847                                    mp->b_cont = tcsr_mp = tpi_ack_alloc(mp->b_cont,
1848                                        sizeof (struct T_capability_ack), M_PCPROTO,
1849                                        T_CAPABILITY_ACK);

1851                                    if (tcsr_mp == NULL) {
1852                                            tilog("timodwproc: allocb failed no "
1853                                                "recovery attempt\n", 0);
1854                                            miocnak(q, mp, 0, ENOMEM);
1855                                            break;
1856                                    }

1858                                    tp->tim_flags &= ~(WAITIOCACK | TI_CAP_RECVD |
1859                                        WAIT_IOCINFOACK | CAP_WANTS_INFO);
1860                                    ((struct T_capability_ack *)
1861                                        tcsr_mp->b_rptr)->CAP_bits1 = 0;
1862                                    tim_ioctl_send_reply(q, mp, tcsr_mp);

1864                                    /*
1865                                     * It could happen when timod is awaiting ack
1866                                     * for TI_GETPEERNAME/TI_GETMYNAME.
1867                                     */
1868                                    if (tp->tim_iocsave != NULL) {
1869                                            freemsg(tp->tim_iocsave);
1870                                            tp->tim_iocsave = NULL;
1871                                            tp->tim_saved_prim = -1;
1872                                    }
1873                                    break;

1875                            default:
1876                                    cmn_err(CE_PANIC,
1877                                        "timodwproc: unknown tpi_capability value "
1878                                        "%d\n", tp->tim_provinfo->tpi_capability);
```



```
1879                                            break;
1880                                    }
1881                            }
1882                            break;

1922 /* ONC_PLUS EXTRACT START */
1884                            case TI_GETMYNAME:

1886                                    tilog("timodwproc: Got TI_GETMYNAME\n", 0);

1888                                    if (tp->tim_provinfo->tpi_myname == PI_YES) {
1889                                            putnext(q, mp);
1890                                            break;
1891                                    }
1892                                    goto getname;

1894                            case TI_GETPEERNAME:

1896                                    tilog("timodwproc: Got TI_GETPEERNAME\n", 0);

1898                                    if (tp->tim_provinfo->tpi_peername == PI_YES) {
1899                                            putnext(q, mp);
1900                                            break;
1901                                    }
1902 getname:
1903                                    if ((tmp = copymsg(mp)) == NULL) {
1904                                            tim_recover(q, mp, msgsize(mp));
1905                                            return (1);
1906                                    }
1907                                    /*
1908                                     * tim_iocsave may be non-NULL when timod is awaiting
1909                                     * ack for another TI_GETPEERNAME/TI_GETMYNAME.
1910                                     */
1911                                    freemsg(tp->tim_iocsave);
1912                                    tp->tim_iocsave = mp;
1913                                    tp->tim_saved_prim = -1;
1914                                    putnext(q, tmp);
1915                                    break;
1916                                    }
1917                            break;

1919                    case M_IOCDATA:

1921                            if (tp->tim_flags & NAMEPROC) {
1922                                    if (ti_doname(q, mp) != DONAME_CONT) {
1923                                            tp->tim_flags &= ~NAMEPROC;
1924                                    }
1925                            } else
1926                                    putnext(q, mp);
1927                            break;

1929                    case M_PROTO:
1930                    case M_PCPROTO:
1931                            if (MBLKL(mp) < sizeof (t_scalar_t)) {
1932                                    merror(q, mp, EPROTO);
1933                                    return (1);
1934                            }

1936                            pptr = (union T_primitives *)mp->b_rptr;
1937                            switch (pptr->type) {
1938                            default:
1939                                    putnext(q, mp);
1940                                    break;

1942                            case T_EXDATA_REQ:
1943                            case T_DATA_REQ:
```

```
1944                        if (pptr->type == T_EXDATA_REQ)
1945                                tilog("timodwproc: Got T_EXDATA_REQ\n", 0);

1947                if (!bcanputnext(q, mp->b_band)) {
1948                        (void) putbq(q, mp);
1949                        return (1);
1950                }
1951                putnext(q, mp);
1952                break;
1992 /* ONC_PLUS EXTRACT END */

1954                case T_UNITDATA_REQ:
1955                        if (tp->tim_flags & CLTS) {
1956                                tmp = tim_filladdr(q, mp, B_TRUE);
1957                                if (tmp == NULL) {
1958                                        return (1);
1959                                } else {
1960                                        mp = tmp;
1961                                }
1962                        }
1963                        if (auditing)
1964                                audit_sock(T_UNITDATA_REQ, q, mp, TIMOD_ID);
1965                if (!bcanputnext(q, mp->b_band)) {
1966                        (void) putbq(q, mp);
1967                        return (1);
1968                }
1969                putnext(q, mp);
1970                break;

2012 /* ONC_PLUS EXTRACT START */
1972                case T_CONN_REQ: {
1973                        struct T_conn_req *reqp = (struct T_conn_req *)
1974                            mp->b_rptr;
1975                        void *p;

1977                        tilog("timodwproc: Got T_CONN_REQ\n", 0);

1979                        if (MBLKL(mp) < sizeof (struct T_conn_req)) {
1980                                merror(q, mp, EPROTO);
1981                                return (1);
1982                        }

1984                        if (tp->tim_flags & DO_PEERNAME) {
1985                                if (!MBLKIN(mp, reqp->DEST_offset,
1986                                    reqp->DEST_length)) {
1987                                        merror(q, mp, EPROTO);
1988                                        return (1);
1989                                }
1990                                ASSERT(reqp->DEST_length >= 0);
1991                                mutex_enter(&tp->tim_mutex);
1992                                if (reqp->DEST_length > tp->tim_peermaxlen) {
1993                                        p = kmem_alloc(reqp->DEST_length,
1994                                            KM_NOSLEEP);
1995                                        if (p == NULL) {
1996                                                mutex_exit(&tp->tim_mutex);
1997                                                tilog("timodwproc: kmem_alloc "
1998                                                    "failed, attempting "
1999                                                    "recovery\n", 0);
2000                                                tim_recover(q, mp,
2001                                                    reqp->DEST_length);
2002                                                return (1);
2003                                        }
2004                                        if (tp->tim_peermaxlen)
2005                                                kmem_free(tp->tim_peername,
2006                                                    tp->tim_peermaxlen);
2007                                        tp->tim_peername = p;
```



```
2008                                        tp->tim_peermaxlen = reqp->DEST_length;
2009                                }
2010                                tp->tim_peerlen = reqp->DEST_length;
2011                                p = mp->b_rptr + reqp->DEST_offset;
2012                                bcopy(p, tp->tim_peername, tp->tim_peerlen);
2013                                mutex_exit(&tp->tim_mutex);
2014                        }
2015                        if (tp->tim_flags & COTS)
2016                                tp->tim_flags |= CONNWAIT;
2058 /* ONC_PLUS EXTRACT END */
2017                        if (auditing)
2018                                audit_sock(T_CONN_REQ, q, mp, TIMOD_ID);
2061 /* ONC_PLUS EXTRACT START */
2019                putnext(q, mp);
2020                break;
2021                }

2023                case O_T_CONN_RES:
2024                case T_CONN_RES: {
2025                        struct T_conn_res *resp;
2026                        struct T_conn_ind *indp;
2027                        mblk_t *pmp = NULL;
2028                        mblk_t *nbp;

2030                        if (MBLKL(mp) < sizeof (struct T_conn_res) ||
2031                            (tp->tim_flags & WAITIOCACK)) {
2032                                merror(q, mp, EPROTO);
2033                                return (1);
2034                        }

2036                        resp = (struct T_conn_res *)mp->b_rptr;
2037                        for (tmp = tp->tim_consave; tmp != NULL;
2038                            tmp = tmp->b_next) {
2039                                indp = (struct T_conn_ind *)tmp->b_rptr;
2040                                if (indp->SEQ_number == resp->SEQ_number)
2041                                        break;
2042                                pmp = tmp;
2043                        }
2044                        if (tmp == NULL)
2045                                goto cresout;

2047                        if ((nbp = dupb(mp)) == NULL &&
2048                            (nbp = copyb(mp)) == NULL) {
2049                                tim_recover(q, mp, msgsize(mp));
2050                                return (1);
2051                        }

2053                        if (pmp != NULL)
2054                                pmp->b_next = tmp->b_next;
2055                        else
2056                                tp->tim_consave = tmp->b_next;
2057                        tmp->b_next = NULL;

2059                        /*
2060                         * Construct a list with:
2061                         *      nbp - copy of user's original request
2062                         *      tmp - the extracted T_conn_ind
2063                         */
2064                        nbp->b_cont = tmp;
2065                        /*
2066                         * tim_iocsave may be non-NULL when timod is awaiting
2067                         * ack for TI_GETPEERNAME/TI_GETMYNAME.
2068                         */
2069                        freemsg(tp->tim_iocsave);
2070                        tp->tim_iocsave = nbp;
2071                        tp->tim_saved_prim = pptr->type;
```

```
2072                        tp->tim_flags |= WAIT_CONNRESACK | WAITIOCACK;

2074                cresout:
2075                        putnext(q, mp);
2076                        break;
2077                }

2122 /* ONC_PLUS EXTRACT END */
2079                case T_DISCON_REQ: {
2080                        struct T_discon_req *disp;
2081                        struct T_conn_ind *conp;
2082                        mblk_t *pmp = NULL;

2084                        if (MBLKL(mp) < sizeof (struct T_discon_req)) {
2085                                merror(q, mp, EPROTO);
2086                                return (1);
2087                        }

2089                        disp = (struct T_discon_req *)mp->b_rptr;
2090                        tp->tim_flags &= ~(CONNWAIT|LOCORDREL|REMORDREL);
2091                        tim_clear_peer(tp);

2093                        /*
2094                         * If we are already connected, there won't
2095                         * be any messages on tim_consave.
2096                         */
2097                        for (tmp = tp->tim_consave; tmp; tmp = tmp->b_next) {
2098                                conp = (struct T_conn_ind *)tmp->b_rptr;
2099                                if (conp->SEQ_number == disp->SEQ_number)
2100                                        break;
2101                                pmp = tmp;
2102                        }
2103                        if (tmp) {
2104                                if (pmp)
2105                                        pmp->b_next = tmp->b_next;
2106                                else
2107                                        tp->tim_consave = tmp->b_next;
2108                                tmp->b_next = NULL;
2109                                freemsg(tmp);
2110                        }
2111                        putnext(q, mp);
2112                        break;
2113                }

2115                case T_ORDREL_REQ:
2116                        if (tp->tim_flags & REMORDREL) {
2117                                tp->tim_flags &= ~(LOCORDREL|REMORDREL);
2118                                tim_clear_peer(tp);
2119                        } else {
2120                                tp->tim_flags |= LOCORDREL;
2121                        }
2122                        putnext(q, mp);
2123                        break;

2125                case T_CAPABILITY_REQ:
2126                        tilog("timodwproc: Got T_CAPABILITY_REQ\n", 0);
2127                        /*
2128                         * XXX: We may know at this point whether transport
2129                         * provides T_CAPABILITY_REQ or not and we may utilise
2130                         * this knowledge here.
2131                         */
2132                        putnext(q, mp);
2133                        break;
2178 /* ONC_PLUS EXTRACT START */
2134                }
2135                break;
```



```
2136        case M_FLUSH:

2138                tilog("timodwproc: Got M_FLUSH\n", 0);

2140                if (*mp->b_rptr & FLUSHW) {
2141                        if (*mp->b_rptr & FLUSHBAND)
2142                                flushband(q, *(mp->b_rptr + 1), FLUSHDATA);
2143                        else
2144                                flushq(q, FLUSHDATA);
2145                }
2146                putnext(q, mp);
2147                break;
2148        }

2150        return (0);
2151 }
______unchanged_portion_omitted_

2449 /* ONC_PLUS EXTRACT END */

2405 /*
2406  * Fill in the address of a connectionless data packet if a connect
2407  * had been done on this endpoint.
2408  */
2409 static mblk_t *
2410 tim_filladdr(queue_t *q, mblk_t *mp, boolean_t dorecover)
2411 {
2412        mblk_t *bp;
2413        struct tim_tim *tp;
2414        struct T_unitdata_req *up;
2415        struct T_unitdata_req *nup;
2416        size_t plen;

2418        tp = (struct tim_tim *)q->q_ptr;
2419        if (mp->b_datap->db_type == M_DATA) {
2420                mutex_enter(&tp->tim_mutex);
2421                bp = allocb(sizeof (struct T_unitdata_req) + tp->tim_peerlen,
2422                    BPRI_MED);
2423                if (bp != NULL) {
2424                        bp->b_datap->db_type = M_PROTO;
2425                        up = (struct T_unitdata_req *)bp->b_rptr;
2426                        up->PRIM_type = T_UNITDATA_REQ;
2427                        up->DEST_length = tp->tim_peerlen;
2428                        bp->b_wptr += sizeof (struct T_unitdata_req);
2429                        up->DEST_offset = sizeof (struct T_unitdata_req);
2430                        up->OPT_length = 0;
2431                        up->OPT_offset = 0;
2432                        if (tp->tim_peerlen > 0) {
2433                                bcopy(tp->tim_peername, bp->b_wptr,
2434                                    tp->tim_peerlen);
2435                                bp->b_wptr += tp->tim_peerlen;
2436                        }
2437                        bp->b_cont = mp;
2438                }
2439        } else {
2440                ASSERT(mp->b_datap->db_type == M_PROTO);
2441                up = (struct T_unitdata_req *)mp->b_rptr;
2442                ASSERT(up->PRIM_type == T_UNITDATA_REQ);
2443                if (up->DEST_length != 0)
2444                        return (mp);
2445                mutex_enter(&tp->tim_mutex);
2446                bp = allocb(sizeof (struct T_unitdata_req) + up->OPT_length +
2447                    tp->tim_peerlen, BPRI_MED);
2448                if (bp != NULL) {
2449                        bp->b_datap->db_type = M_PROTO;
2450                        nup = (struct T_unitdata_req *)bp->b_rptr;
```

```
2451                        nup->PRIM_type = T_UNITDATA_REQ;
2452                        nup->DEST_length = plen = tp->tim_peerlen;
2453                        bp->b_wptr += sizeof (struct T_unitdata_req);
2454                        nup->DEST_offset = sizeof (struct T_unitdata_req);
2455                        if (plen > 0) {
2456                                bcopy(tp->tim_peername, bp->b_wptr, plen);
2457                                bp->b_wptr += plen;
2458                        }
2459                        mutex_exit(&tp->tim_mutex);
2460                        if (up->OPT_length == 0) {
2461                                nup->OPT_length = 0;
2462                                nup->OPT_offset = 0;
2463                        } else {
2464                                nup->OPT_length = up->OPT_length;
2465                                nup->OPT_offset =
2466                                    sizeof (struct T_unitdata_req) + plen;
2467                                bcopy((mp->b_wptr + up->OPT_offset), bp->b_wptr,
2468                                    up->OPT_length);
2469                                bp->b_wptr += up->OPT_length;
2470                        }
2471                        bp->b_cont = mp->b_cont;
2472                        mp->b_cont = NULL;
2473                        freeb(mp);
2474                        return (bp);
2475                }
2476        }
2477        ASSERT(MUTEX_HELD(&tp->tim_mutex));
2478        if (bp == NULL && dorecover) {
2479                tim_recover(q, mp,
2480                    sizeof (struct T_unitdata_req) + tp->tim_peerlen);
2481        }
2482        mutex_exit(&tp->tim_mutex);
2483        return (bp);
2484 }
```

_____***unchanged_portion_omitted_***

```
2583 /* ONC_PLUS EXTRACT START */
2537 static void
2538 tim_recover(queue_t *q, mblk_t *mp, t_scalar_t size)
2539 {
2540        struct tim_tim  *tp;
2541        bufcall_id_t    bid;
2542        timeout_id_t    tid;

2544        tp = (struct tim_tim *)q->q_ptr;

2546        /*
2547         * Avoid re-enabling the queue.
2548         */
2549        if (mp->b_datap->db_type == M_PCPROTO)
2550                mp->b_datap->db_type = M_PROTO;
2551        noenable(q);
2552        (void) putbq(q, mp);

2554        /*
2555         * Make sure there is at most one outstanding request per queue.
2556         */
2557        if (q->q_flag & QREADR) {
2558                if (tp->tim_rtimoutid || tp->tim_rbufcid)
2559                        return;
2560        } else {
2561                if (tp->tim_wtimoutid || tp->tim_wbufcid)
2562                        return;
2563        }
2564        if (!(bid = qbufcall(RD(q), (size_t)size, BPRI_MED, tim_buffer, q))) {
2565                tid = qtimeout(RD(q), tim_timer, q, TIMWAIT);
```



```
2566                if (q->q_flag & QREADR)
2567                        tp->tim_rtimoutid = tid;
2568                else
2569                        tp->tim_wtimoutid = tid;
2570        } else  {
2571                if (q->q_flag & QREADR)
2572                        tp->tim_rbufcid = bid;
2573                else
2574                        tp->tim_wbufcid = bid;
2575        }
2576 }
```

_____***unchanged_portion_omitted_***

```
2694 /* ONC_PLUS EXTRACT END */
2647 static void
2648 tim_tcap_timer(void *q_ptr)
2649 {
2650        queue_t *q = (queue_t *)q_ptr;
2651        struct tim_tim *tp = (struct tim_tim *)q->q_ptr;

2653        ASSERT(tp != NULL && tp->tim_tcap_timoutid != 0);
2654        ASSERT((tp->tim_flags & TI_CAP_RECVD) != 0);

2656        tp->tim_tcap_timoutid = 0;
2657        TILOG("tim_tcap_timer: fired\n", 0);
2658        tim_tcap_genreply(q, tp);
2659 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
  106185 Thu Jul 11 01:30:06 2013
new/usr/src/uts/common/os/flock.c
onc plus-be-gone
**********************************************************
```

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

  22 /*
  23  * Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
  24  * Use is subject to license terms.
  25  */

  27 /*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
  28 /*      All Rights Reserved */

  30 #pragma ident   "%Z%%M% %I%     %E% SMI"

  32 #include <sys/flock_impl.h>
  33 #include <sys/vfs.h>
  34 #include <sys/t_lock.h>         /* for <sys/callb.h> */
  35 #include <sys/callb.h>
  36 #include <sys/clconf.h>
  37 #include <sys/cladm.h>
  38 #include <sys/nbmlock.h>
  39 #include <sys/cred.h>
  40 #include <sys/policy.h>

  42 /*
  43  * The following four variables are for statistics purposes and they are
  44  * not protected by locks. They may not be accurate but will at least be
  45  * close to the actual value.
  46  */

  48 int     flk_lock_allocs;
  49 int     flk_lock_frees;
  50 int     edge_allocs;
  51 int     edge_frees;
  52 int     flk_proc_vertex_allocs;
  53 int     flk_proc_edge_allocs;
  54 int     flk_proc_vertex_frees;
  55 int     flk_proc_edge_frees;

  57 static kmutex_t flock_lock;

  59 #ifdef DEBUG
  60 int check_debug = 0;
```



```
  61 #define CHECK_ACTIVE_LOCKS(gp)  if (check_debug) \
  62                                         check_active_locks(gp);
  63 #define CHECK_SLEEPING_LOCKS(gp)        if (check_debug) \
  64                                                 check_sleeping_locks(gp);
  65 #define CHECK_OWNER_LOCKS(gp, pid, sysid, vp)   \
  66                 if (check_debug)        \
  67                         check_owner_locks(gp, pid, sysid, vp);
  68 #define CHECK_LOCK_TRANSITION(old_state, new_state) \
  69         { \
  70                 if (check_lock_transition(old_state, new_state)) { \
  71                         cmn_err(CE_PANIC, "Illegal lock transition \
  72                             from %d to %d", old_state, new_state); \
  73                 } \
  74         }
  75 #else

  77 #define CHECK_ACTIVE_LOCKS(gp)
  78 #define CHECK_SLEEPING_LOCKS(gp)
  79 #define CHECK_OWNER_LOCKS(gp, pid, sysid, vp)
  80 #define CHECK_LOCK_TRANSITION(old_state, new_state)

  82 #endif /* DEBUG */

  84 struct kmem_cache       *flk_edge_cache;

  86 graph_t         *lock_graph[HASH_SIZE];
  87 proc_graph_t    pgraph;

  89 /*
  90  * Clustering.
  91  *
  92  * NLM REGISTRY TYPE IMPLEMENTATION
  93  *
  94  * Assumptions:
  95  *  1.  Nodes in a cluster are numbered starting at 1; always non-negative
  96  *      integers; maximum node id is returned by clconf_maximum_nodeid().
  97  *  2.  We use this node id to identify the node an NLM server runs on.
  98  */

 100 /*
 101  * NLM registry object keeps track of NLM servers via their
 102  * nlmids (which are the node ids of the node in the cluster they run on)
 103  * that have requested locks at this LLM with which this registry is
 104  * associated.
 105  *
 106  * Representation of abstraction:
 107  *    rep = record[     states: array[nlm_state],
 108  *                      lock: mutex]
 109  *
 110  *    Representation invariants:
 111  *      1. index i of rep.states is between 0 and n - 1 where n is number
 112  *         of elements in the array, which happen to be the maximum number
 113  *         of nodes in the cluster configuration + 1.
 114  *      2. map nlmid to index i of rep.states
 115  *              0   -> 0
 116  *              1   -> 1
 117  *              2   -> 2
 118  *              n-1 -> clconf_maximum_nodeid()+1
 119  *      3.  This 1-1 mapping is quite convenient and it avoids errors resulting
 120  *          from forgetting to subtract 1 from the index.
 121  *      4.  The reason we keep the 0th index is the following.  A legitimate
 122  *          cluster configuration includes making a UFS file system NFS
 123  *          exportable.  The code is structured so that if you're in a cluster
 124  *          you do one thing; otherwise, you do something else.  The problem
 125  *          is what to do if you think you're in a cluster with PXFS loaded,
 126  *          but you're using UFS not PXFS?  The upper two bytes of the sysid
```

```
 127  *          encode the node id of the node where NLM server runs; these bytes
 128  *          are zero for UFS.  Since the nodeid is used to index into the
 129  *          registry, we can record the NLM server state information at index
 130  *          0 using the same mechanism used for PXFS file locks!
 131  */
 132 static flk_nlm_status_t *nlm_reg_status = NULL; /* state array 0..N-1 */
 133 static kmutex_t nlm_reg_lock;                   /* lock to protect arrary */
 134 static uint_t nlm_status_size;                  /* size of state array */

 136 /*
 137  * Although we need a global lock dependency graph (and associated data
 138  * structures), we also need a per-zone notion of whether the lock manager is
 139  * running, and so whether to allow lock manager requests or not.
 140  *
 141  * Thus, on a per-zone basis we maintain a ''global'' variable
 142  * (flk_lockmgr_status), protected by flock_lock, and set when the lock
 143  * manager is determined to be changing state (starting or stopping).
 144  *
 145  * Each graph/zone pair also has a copy of this variable, which is protected by
 146  * the graph's mutex.
 147  *
 148  * The per-graph copies are used to synchronize lock requests with shutdown
 149  * requests.  The global copy is used to initialize the per-graph field when a
 150  * new graph is created.
 151  */
 152 struct flock_globals {
 153         flk_lockmgr_status_t flk_lockmgr_status;
 154         flk_lockmgr_status_t lockmgr_status[HASH_SIZE];
 155 };
_____unchanged_portion_omitted_
 612 /* ONC_PLUS EXTRACT END */

 612 /*
 613  * Initialize the flk_edge_cache data structure and create the
 614  * nlm_reg_status array.
 615  */

 617 void
 618 flk_init(void)
 619 {
 620         uint_t  i;

 622         flk_edge_cache = kmem_cache_create("flk_edges",
 623                 sizeof (struct edge), 0, NULL, NULL, NULL, NULL, NULL, 0);
 624         if (flk_edge_cache == NULL) {
 625                 cmn_err(CE_PANIC, "Couldn't create flk_edge_cache\n");
 626         }
 627         /*
 628          * Create the NLM registry object.
 629          */

 631         if (cluster_bootflags & CLUSTER_BOOTED) {
 632                 /*
 633                  * This routine tells you the maximum node id that will be used
 634                  * in the cluster.  This number will be the size of the nlm
 635                  * registry status array.  We add 1 because we will be using
 636                  * all entries indexed from 0 to maxnodeid; e.g., from 0
 637                  * to 64, for a total of 65 entries.
 638                  */
 639                 nlm_status_size = clconf_maximum_nodeid() + 1;
 640         } else {
 641                 nlm_status_size = 0;
 642         }

 644         if (nlm_status_size != 0) {     /* booted as a cluster */
 645                 nlm_reg_status = (flk_nlm_status_t *)
```



```
 646                         kmem_alloc(sizeof (flk_nlm_status_t) * nlm_status_size,
 647                                 KM_SLEEP);

 649                 /* initialize all NLM states in array to NLM_UNKNOWN */
 650                 for (i = 0; i < nlm_status_size; i++) {
 651                         nlm_reg_status[i] = FLK_NLM_UNKNOWN;
 652                 }
 653         }
 654 }
_____unchanged_portion_omitted_

 987 /* ONC_PLUS EXTRACT START */
 985 /*
 986  * The actual execution of the request in the simple case is only to
 987  * insert the 'request' in the list of active locks if it is not an
 988  * UNLOCK.
 989  * We have to consider the existing active locks' relation to
 990  * this 'request' if they are owned by same process. flk_relation() does
 991  * this job and sees to that the dependency graph information is maintained
 992  * properly.
 993  */

 995 int
 996 flk_execute_request(lock_descriptor_t *request)
 997 {
 998         graph_t *gp = request->l_graph;
 999         vnode_t *vp = request->l_vnode;
1000         lock_descriptor_t       *lock, *lock1;
1001         int done_searching = 0;

1003         CHECK_SLEEPING_LOCKS(gp);
1004         CHECK_ACTIVE_LOCKS(gp);

1006         ASSERT(MUTEX_HELD(&gp->gp_mutex));

1008         flk_set_state(request, FLK_START_STATE);

1010         ASSERT(NOT_BLOCKED(request));

1012         /* IO_LOCK requests are only to check status */

1014         if (IS_IO_LOCK(request))
1015                 return (0);

1017         SET_LOCK_TO_FIRST_ACTIVE_VP(gp, lock, vp);

1019         if (lock == NULL && request->l_type == F_UNLCK)
1020                 return (0);
1021         if (lock == NULL) {
1022                 flk_insert_active_lock(request);
1023                 return (0);
1024         }

1026         do {
1027                 lock1 = lock->l_next;
1028                 if (SAME_OWNER(request, lock)) {
1029                         done_searching = flk_relation(lock, request);
1030                 }
1031                 lock = lock1;
1032         } while (lock->l_vnode == vp && !done_searching);

1034         /*
1035          * insert in active queue
1036          */

1038         if (request->l_type != F_UNLCK)
```

```
1039                flk_insert_active_lock(request);

1041        return (0);
1042 }
1046 /* ONC_PLUS EXTRACT END */

1044 /*
1045  * 'request' is blocked by some one therefore we put it into sleep queue.
1046  */
1047 static int
1048 flk_wait_execute_request(lock_descriptor_t *request)
1049 {
1050        graph_t *gp = request->l_graph;
1051        callb_cpr_t     *cprp;          /* CPR info from callback */
1052        struct flock_globals *fg;
1053        int index;

1055        ASSERT(MUTEX_HELD(&gp->gp_mutex));
1056        ASSERT(IS_WILLING_TO_SLEEP(request));

1058        flk_insert_sleeping_lock(request);

1060        if (IS_LOCKMGR(request)) {
1061                index = HASH_INDEX(request->l_vnode);
1062                fg = flk_get_globals();

1064                if (nlm_status_size == 0) {     /* not booted as a cluster */
1065                        if (fg->lockmgr_status[index] != FLK_LOCKMGR_UP) {
1066                                flk_cancel_sleeping_lock(request, 1);
1067                                return (ENOLCK);
1068                        }
1069                } else {                        /* booted as a cluster */
1070                        /*
1071                         * If the request is an NLM server lock request,
1072                         * and the NLM state of the lock request is not
1073                         * NLM_UP (because the NLM server is shutting
1074                         * down), then cancel the sleeping lock and
1075                         * return error ENOLCK that will encourage the
1076                         * client to retransmit.
1077                         */
1078                        if (!IS_NLM_UP(request)) {
1079                                flk_cancel_sleeping_lock(request, 1);
1080                                return (ENOLCK);
1081                        }
1082                }
1083        }

1085        /* Clustering: For blocking PXFS locks, return */
1086        if (IS_PXFS(request)) {
1087                /*
1088                 * PXFS locks sleep on the client side.
1089                 * The callback argument is used to wake up the sleeper
1090                 * when the lock is granted.
1091                 * We return -1 (rather than an errno value) to indicate
1092                 * the client side should sleep
1093                 */
1094                return (PXFS_LOCK_BLOCKED);
1095        }

1097        if (request->l_callbacks != NULL) {
1098                /*
1099                 * To make sure the shutdown code works correctly, either
1100                 * the callback must happen after putting the lock on the
1101                 * sleep list, or we must check the shutdown status after
1102                 * returning from the callback (and before sleeping).  At
1103                 * least for now, we'll use the first option.  If a
```



```
1104                 * shutdown or signal or whatever happened while the graph
1105                 * mutex was dropped, that will be detected by
1106                 * wait_for_lock().
1107                 */
1108                mutex_exit(&gp->gp_mutex);

1110                cprp = flk_invoke_callbacks(request->l_callbacks,
1111                                            FLK_BEFORE_SLEEP);

1113                mutex_enter(&gp->gp_mutex);

1115                if (cprp == NULL) {
1116                        wait_for_lock(request);
1117                } else {
1118                        mutex_enter(cprp->cc_lockp);
1119                        CALLB_CPR_SAFE_BEGIN(cprp);
1120                        mutex_exit(cprp->cc_lockp);
1121                        wait_for_lock(request);
1122                        mutex_enter(cprp->cc_lockp);
1123                        CALLB_CPR_SAFE_END(cprp, cprp->cc_lockp);
1124                        mutex_exit(cprp->cc_lockp);
1125                }

1127                mutex_exit(&gp->gp_mutex);
1128                (void) flk_invoke_callbacks(request->l_callbacks,
1129                                            FLK_AFTER_SLEEP);
1130                mutex_enter(&gp->gp_mutex);
1131        } else {
1132                wait_for_lock(request);
1133        }

1135        if (IS_LOCKMGR(request)) {
1136                /*
1137                 * If the lock manager is shutting down, return an
1138                 * error that will encourage the client to retransmit.
1139                 */
1140                if (fg->lockmgr_status[index] != FLK_LOCKMGR_UP &&
1141                        !IS_GRANTED(request)) {
1142                        flk_cancel_sleeping_lock(request, 1);
1143                        return (ENOLCK);
1144                }
1145        }

1147        if (IS_INTERRUPTED(request)) {
1148                /* we got a signal, or act like we did */
1149                flk_cancel_sleeping_lock(request, 1);
1150                return (EINTR);
1151        }

1153        /* Cancelled if some other thread has closed the file */

1155        if (IS_CANCELLED(request)) {
1156                flk_cancel_sleeping_lock(request, 1);
1157                return (EBADF);
1158        }

1160        request->l_state &= ~GRANTED_LOCK;
1161        REMOVE_SLEEP_QUEUE(request);
1162        return (flk_execute_request(request));
1163 }
```

unchanged_portion_omitted_

```
2246 /* ONC_PLUS EXTRACT START */
2242 /*
2243  * Determine whether there are any locks for the given vnode with a remote
2244  * sysid.  Returns zero if not, non-zero if there are.
```

```
2245  *
2246  * Note that the return value from this function is potentially invalid
2247  * once it has been returned.  The caller is responsible for providing its
2248  * own synchronization mechanism to ensure that the return value is useful
2249  * (e.g., see nfs_lockcompletion()).
2250  */
2251 int
2252 flk_has_remote_locks(vnode_t *vp)
2253 {
2254         lock_descriptor_t *lock;
2255         int result = 0;
2256         graph_t *gp;

2258         gp = flk_get_lock_graph(vp, FLK_USE_GRAPH);
2259         if (gp == NULL) {
2260                 return (0);
2261         }

2263         mutex_enter(&gp->gp_mutex);

2265         SET_LOCK_TO_FIRST_ACTIVE_VP(gp, lock, vp);

2267         if (lock) {
2268                 while (lock->l_vnode == vp) {
2269                         if (IS_REMOTE(lock)) {
2270                                 result = 1;
2271                                 goto done;
2272                         }
2273                         lock = lock->l_next;
2274                 }
2275         }

2277         SET_LOCK_TO_FIRST_SLEEP_VP(gp, lock, vp);

2279         if (lock) {
2280                 while (lock->l_vnode == vp) {
2281                         if (IS_REMOTE(lock)) {
2282                                 result = 1;
2283                                 goto done;
2284                         }
2285                         lock = lock->l_next;
2286                 }
2287         }

2289 done:
2290         mutex_exit(&gp->gp_mutex);
2291         return (result);
2292 }
______unchanged_portion_omitted_
2617 /* ONC_PLUS EXTRACT END */


2614 /*
2615  * Called from 'fs' read and write routines for files that have mandatory
2616  * locking enabled.
2617  */

2619 int
2620 chklock(
2621         struct vnode    *vp,
2622         int             iomode,
2623         u_offset_t      offset,
2624         ssize_t         len,
2625         int             fmode,
2626         caller_context_t *ct)
2627 {
```



```
2628         register int    i;
2629         struct flock64  bf;
2630         int             error = 0;

2632         bf.l_type = (iomode & FWRITE) ? F_WRLCK : F_RDLCK;
2633         bf.l_whence = 0;
2634         bf.l_start = offset;
2635         bf.l_len = len;
2636         if (ct == NULL) {
2637                 bf.l_pid = curproc->p_pid;
2638                 bf.l_sysid = 0;
2639         } else {
2640                 bf.l_pid = ct->cc_pid;
2641                 bf.l_sysid = ct->cc_sysid;
2642         }
2643         i = (fmode & (FNDELAY|FNONBLOCK)) ? INOFLCK : INOFLCK|SLPFLCK;
2644         if ((i = reclock(vp, &bf, i, 0, offset, NULL)) != 0 ||
2645             bf.l_type != F_UNLCK)
2646                 error = i ? i : EAGAIN;
2647         return (error);
2648 }

2656 /* ONC_PLUS EXTRACT START */
2650 /*
2651  * convoff - converts the given data (start, whence) to the
2652  * given whence.
2653  */
2654 int
2655 convoff(vp, lckdat, whence, offset)
2656         struct vnode    *vp;
2657         struct flock64  *lckdat;
2658         int             whence;
2659         offset_t        offset;
2660 {
2661         int             error;
2662         struct vattr    vattr;

2664         if ((lckdat->l_whence == 2) || (whence == 2)) {
2665                 vattr.va_mask = AT_SIZE;
2666                 if (error = VOP_GETATTR(vp, &vattr, 0, CRED(), NULL))
2667                         return (error);
2668         }

2670         switch (lckdat->l_whence) {
2671         case 1:
2672                 lckdat->l_start += offset;
2673                 break;
2674         case 2:
2675                 lckdat->l_start += vattr.va_size;
2676                 /* FALLTHRU */
2677         case 0:
2678                 break;
2679         default:
2680                 return (EINVAL);
2681         }

2683         if (lckdat->l_start < 0)
2684                 return (EINVAL);

2686         switch (whence) {
2687         case 1:
2688                 lckdat->l_start -= offset;
2689                 break;
2690         case 2:
2691                 lckdat->l_start -= vattr.va_size;
2692                 /* FALLTHRU */
```

```
2693         case 0:
2694                 break;
2695         default:
2696                 return (EINVAL);
2697         }

2699         lckdat->l_whence = (short)whence;
2700         return (0);
2701 }
2709 /* ONC_PLUS EXTRACT END */


2704 /*      proc_graph function definitions */

2706 /*
2707  * Function checks for deadlock due to the new 'lock'. If deadlock found
2708  * edges of this lock are freed and returned.
2709  */

2711 static int
2712 flk_check_deadlock(lock_descriptor_t *lock)
2713 {
2714         proc_vertex_t   *start_vertex, *pvertex;
2715         proc_vertex_t *dvertex;
2716         proc_edge_t *pep, *ppep;
2717         edge_t  *ep, *nep;
2718         proc_vertex_t *process_stack;

2720         STACK_INIT(process_stack);

2722         mutex_enter(&flock_lock);
2723         start_vertex = flk_get_proc_vertex(lock);
2724         ASSERT(start_vertex != NULL);

2726         /* construct the edges from this process to other processes */

2728         ep = FIRST_ADJ(lock);
2729         while (ep != HEAD(lock)) {
2730                 proc_vertex_t *adj_proc;

2732                 adj_proc = flk_get_proc_vertex(ep->to_vertex);
2733                 for (pep = start_vertex->edge; pep != NULL; pep = pep->next) {
2734                         if (pep->to_proc == adj_proc) {
2735                                 ASSERT(pep->refcount);
2736                                 pep->refcount++;
2737                                 break;
2738                         }
2739                 }
2740                 if (pep == NULL) {
2741                         pep = flk_get_proc_edge();
2742                         pep->to_proc = adj_proc;
2743                         pep->refcount = 1;
2744                         adj_proc->incount++;
2745                         pep->next = start_vertex->edge;
2746                         start_vertex->edge = pep;
2747                 }
2748                 ep = NEXT_ADJ(ep);
2749         }

2751         ep = FIRST_IN(lock);

2753         while (ep != HEAD(lock)) {
2754                 proc_vertex_t *in_proc;

2756                 in_proc = flk_get_proc_vertex(ep->from_vertex);
```



```
2758                 for (pep = in_proc->edge; pep != NULL; pep = pep->next) {
2759                         if (pep->to_proc == start_vertex) {
2760                                 ASSERT(pep->refcount);
2761                                 pep->refcount++;
2762                                 break;
2763                         }
2764                 }
2765                 if (pep == NULL) {
2766                         pep = flk_get_proc_edge();
2767                         pep->to_proc = start_vertex;
2768                         pep->refcount = 1;
2769                         start_vertex->incount++;
2770                         pep->next = in_proc->edge;
2771                         in_proc->edge = pep;
2772                 }
2773                 ep = NEXT_IN(ep);
2774         }

2776         if (start_vertex->incount == 0) {
2777                 mutex_exit(&flock_lock);
2778                 return (0);
2779         }

2781         flk_proc_graph_uncolor();

2783         start_vertex->p_sedge = start_vertex->edge;

2785         STACK_PUSH(process_stack, start_vertex, p_stack);

2787         while ((pvertex = STACK_TOP(process_stack)) != NULL) {
2788                 for (pep = pvertex->p_sedge; pep != NULL; pep = pep->next) {
2789                         dvertex = pep->to_proc;
2790                         if (!PROC_ARRIVED(dvertex)) {
2791                                 STACK_PUSH(process_stack, dvertex, p_stack);
2792                                 dvertex->p_sedge = dvertex->edge;
2793                                 PROC_ARRIVE(pvertex);
2794                                 pvertex->p_sedge = pep->next;
2795                                 break;
2796                         }
2797                         if (!PROC_DEPARTED(dvertex))
2798                                 goto deadlock;
2799                 }
2800                 if (pep == NULL) {
2801                         PROC_DEPART(pvertex);
2802                         STACK_POP(process_stack, p_stack);
2803                 }
2804         }
2805         mutex_exit(&flock_lock);
2806         return (0);

2808 deadlock:

2810         /* we remove all lock edges and proc edges */

2812         ep = FIRST_ADJ(lock);
2813         while (ep != HEAD(lock)) {
2814                 proc_vertex_t *adj_proc;
2815                 adj_proc = flk_get_proc_vertex(ep->to_vertex);
2816                 nep = NEXT_ADJ(ep);
2817                 IN_LIST_REMOVE(ep);
2818                 ADJ_LIST_REMOVE(ep);
2819                 flk_free_edge(ep);
2820                 ppep = start_vertex->edge;
2821                 for (pep = start_vertex->edge; pep != NULL; ppep = pep,
2822                                                 pep = ppep->next) {
2823                         if (pep->to_proc == adj_proc) {
```

```
2824                                pep->refcount--;
2825                                if (pep->refcount == 0) {
2826                                        if (pep == ppep) {
2827                                                start_vertex->edge = pep->next;
2828                                        } else {
2829                                                ppep->next = pep->next;
2830                                        }
2831                                        adj_proc->incount--;
2832                                        flk_proc_release(adj_proc);
2833                                        flk_free_proc_edge(pep);
2834                                }
2835                                break;
2836                        }
2837                }
2838                ep = nep;
2839        }
2840        ep = FIRST_IN(lock);
2841        while (ep != HEAD(lock)) {
2842                proc_vertex_t *in_proc;
2843                in_proc = flk_get_proc_vertex(ep->from_vertex);
2844                nep = NEXT_IN(ep);
2845                IN_LIST_REMOVE(ep);
2846                ADJ_LIST_REMOVE(ep);
2847                flk_free_edge(ep);
2848                ppep = in_proc->edge;
2849                for (pep = in_proc->edge; pep != NULL; ppep = pep,
2850                                        pep = ppep->next) {
2851                        if (pep->to_proc == start_vertex) {
2852                                pep->refcount--;
2853                                if (pep->refcount == 0) {
2854                                        if (pep == ppep) {
2855                                                in_proc->edge = pep->next;
2856                                        } else {
2857                                                ppep->next = pep->next;
2858                                        }
2859                                        start_vertex->incount--;
2860                                        flk_proc_release(in_proc);
2861                                        flk_free_proc_edge(pep);
2862                                }
2863                                break;
2864                        }
2865                }
2866                ep = nep;
2867        }
2868        flk_proc_release(start_vertex);
2869        mutex_exit(&flock_lock);
2870        return (1);
2871 }
```

*unchanged_portion_omitted_*

```
3073 /* ONC_PLUS EXTRACT START */
3065 /*
3066  * Set the control status for lock manager requests.
3067  *
3068  */

3070 /*
3071  * PSARC case 1997/292
3072  *
3073  * Requires: "nlmid" must be >= 1 and <= clconf_maximum_nodeid().
3074  * Effects: Set the state of the NLM server identified by "nlmid"
3075  *   in the NLM registry to state "nlm_state."
3076  *   Raises exception no_such_nlm if "nlmid" doesn't identify a known
3077  *   NLM server to this LLM.
3078  *   Note that when this routine is called with NLM_SHUTTING_DOWN there
3079  *   may be locks requests that have gotten started but not finished.  In
```



```
3080  *   particular, there may be blocking requests that are in the callback code
3081  *   before sleeping (so they're not holding the lock for the graph).  If
3082  *   such a thread reacquires the graph's lock (to go to sleep) after
3083  *   NLM state in the NLM registry  is set to a non-up value,
3084  *   it will notice the status and bail out.  If the request gets
3085  *   granted before the thread can check the NLM registry, let it
3086  *   continue normally.  It will get flushed when we are called with NLM_DOWN.
3087  *
3088  * Modifies: nlm_reg_obj (global)
3089  * Arguments:
3090  *    nlmid     (IN):    id uniquely identifying an NLM server
3091  *    nlm_state (IN):    NLM server state to change "nlmid" to
3092  */
3093 void
3094 cl_flk_set_nlm_status(int nlmid, flk_nlm_status_t nlm_state)
3095 {
3096         /*
3097          * Check to see if node is booted as a cluster. If not, return.
3098          */
3099         if ((cluster_bootflags & CLUSTER_BOOTED) == 0) {
3100                 return;
3101         }

3103         /*
3104          * Check for development/debugging.  It is possible to boot a node
3105          * in non-cluster mode, and then run a special script, currently
3106          * available only to developers, to bring up the node as part of a
3107          * cluster.  The problem is that running such a script does not
3108          * result in the routine flk_init() being called and hence global array
3109          * nlm_reg_status is NULL.  The NLM thinks it's in cluster mode,
3110          * but the LLM needs to do an additional check to see if the global
3111          * array has been created or not. If nlm_reg_status is NULL, then
3112          * return, else continue.
3113          */
3114         if (nlm_reg_status == NULL) {
3115                 return;
3116         }

3118         ASSERT(nlmid <= nlm_status_size && nlmid >= 0);
3119         mutex_enter(&nlm_reg_lock);

3121         if (FLK_REGISTRY_IS_NLM_UNKNOWN(nlm_reg_status, nlmid)) {
3122                 /*
3123                  * If the NLM server "nlmid" is unknown in the NLM registry,
3124                  * add it to the registry in the nlm shutting down state.
3125                  */
3126                 FLK_REGISTRY_CHANGE_NLM_STATE(nlm_reg_status, nlmid,
3127                         FLK_NLM_SHUTTING_DOWN);
3128         } else {
3129                 /*
3130                  * Change the state of the NLM server identified by "nlmid"
3131                  * in the NLM registry to the argument "nlm_state."
3132                  */
3133                 FLK_REGISTRY_CHANGE_NLM_STATE(nlm_reg_status, nlmid,
3134                         nlm_state);
3135         }

3137         /*
3138          *  The reason we must register the NLM server that is shutting down
3139          *  with an LLM that doesn't already know about it (never sent a lock
3140          *  request) is to handle correctly a race between shutdown and a new
3141          *  lock request.  Suppose that a shutdown request from the NLM server
3142          *  invokes this routine at the LLM, and a thread is spawned to
3143          *  service the request. Now suppose a new lock request is in
3144          *  progress and has already passed the first line of defense in
3145          *  reclock(), which denies new locks requests from NLM servers
```

```
3146          *  that are not in the NLM_UP state.  After the current routine
3147          *  is invoked for both phases of shutdown, the routine will return,
3148          *  having done nothing, and the lock request will proceed and
3149          *  probably be granted.  The problem is that the shutdown was ignored
3150          *  by the lock request because there was no record of that NLM server
3151          *  shutting down.   We will be in the peculiar position of thinking
3152          *  that we've shutdown the NLM server and all locks at all LLMs have
3153          *  been discarded, but in fact there's still one lock held.
3154          *  The solution is to record the existence of NLM server and change
3155          *  its state immediately to NLM_SHUTTING_DOWN.  The lock request in
3156          *  progress may proceed because the next phase NLM_DOWN will catch
3157          *  this lock and discard it.
3158          */
3159         mutex_exit(&nlm_reg_lock);

3161         switch (nlm_state) {
3162         case FLK_NLM_UP:
3163                 /*
3164                  * Change the NLM state of all locks still held on behalf of
3165                  * the NLM server identified by "nlmid" to NLM_UP.
3166                  */
3167                 cl_flk_change_nlm_state_all_locks(nlmid, FLK_NLM_UP);
3168                 break;

3170         case FLK_NLM_SHUTTING_DOWN:
3171                 /*
3172                  * Wake up all sleeping locks for the NLM server identified
3173                  * by "nlmid." Note that eventually all woken threads will
3174                  * have their lock requests cancelled and descriptors
3175                  * removed from the sleeping lock list.  Note that the NLM
3176                  * server state associated with each lock descriptor is
3177                  * changed to FLK_NLM_SHUTTING_DOWN.
3178                  */
3179                 cl_flk_wakeup_sleeping_nlm_locks(nlmid);
3180                 break;

3182         case FLK_NLM_DOWN:
3183                 /*
3184                  * Discard all active, granted locks for this NLM server
3185                  * identified by "nlmid."
3186                  */
3187                 cl_flk_unlock_nlm_granted(nlmid);
3188                 break;

3190         default:
3191                 panic("cl_set_nlm_status: bad status (%d)", nlm_state);
3192         }
3193 }
```

_____unchanged_portion_omitted_

```
3695 /* ONC_PLUS EXTRACT END */


3688 /*
3689  * Wait until a lock is granted, cancelled, or interrupted.
3690  */

3692 static void
3693 wait_for_lock(lock_descriptor_t *request)
3694 {
3695         graph_t *gp = request->l_graph;

3697         ASSERT(MUTEX_HELD(&gp->gp_mutex));

3699         while (!(IS_GRANTED(request)) && !(IS_CANCELLED(request)) &&
3700             !(IS_INTERRUPTED(request))) {
3701                 if (!cv_wait_sig(&request->l_cv, &gp->gp_mutex)) {
```



```
3702                         flk_set_state(request, FLK_INTERRUPTED_STATE);
3703                         request->l_state |= INTERRUPTED_LOCK;
3704                 }
3705         }
3706 }
```

```
3718 /* ONC_PLUS EXTRACT START */
3708 /*
3709  * Create an flock structure from the existing lock information
3710  *
3711  * This routine is used to create flock structures for the lock manager
3712  * to use in a reclaim request.  Since the lock was originated on this
3713  * host, it must be conforming to UNIX semantics, so no checking is
3714  * done to make sure it falls within the lower half of the 32-bit range.
3715  */

3717 static void
3718 create_flock(lock_descriptor_t *lp, flock64_t *flp)
3719 {
3720         ASSERT(lp->l_end == MAX_U_OFFSET_T || lp->l_end <= MAXEND);
3721         ASSERT(lp->l_end >= lp->l_start);

3723         flp->l_type = lp->l_type;
3724         flp->l_whence = 0;
3725         flp->l_start = lp->l_start;
3726         flp->l_len = (lp->l_end == MAX_U_OFFSET_T) ? 0 :
3727                 (lp->l_end - lp->l_start + 1);
3728         flp->l_sysid = lp->l_flock.l_sysid;
3729         flp->l_pid = lp->l_flock.l_pid;
3730 }
```

_____unchanged_portion_omitted_

```
3873 /* ONC_PLUS EXTRACT END */

3863 /*
3864  * PSARC case 1997/292
3865  */
3866 /*
3867  * This is the public routine exported by flock.h.
3868  */
3869 void
3870 cl_flk_change_nlm_state_to_unknown(int nlmid)
3871 {
3872         /*
3873          * Check to see if node is booted as a cluster. If not, return.
3874          */
3875         if ((cluster_bootflags & CLUSTER_BOOTED) == 0) {
3876                 return;
3877         }

3879         /*
3880          * See comment in cl_flk_set_nlm_status().
3881          */
3882         if (nlm_reg_status == NULL) {
3883                 return;
3884         }

3886         /*
3887          * protect NLM registry state with a mutex.
3888          */
3889         ASSERT(nlmid <= nlm_status_size && nlmid >= 0);
3890         mutex_enter(&nlm_reg_lock);
3891         FLK_REGISTRY_CHANGE_NLM_STATE(nlm_reg_status, nlmid, FLK_NLM_UNKNOWN);
3892         mutex_exit(&nlm_reg_lock);
3893 }
```

_____unchanged_portion_omitted_

```
4217 #endif /* DEBUG */
```

```
**********************************************************
   73369 Thu Jul 11 01:30:07 2013
new/usr/src/uts/common/os/sig.c
onc plus-be-gone
**********************************************************
______unchanged_portion_omitted_

2230 /* ONC_PLUS EXTRACT START */
2230 void
2231 sigintr(k_sigset_t *smask, int intable)
2232 {
2233         proc_t *p;
2234         int owned;
2235         k_sigset_t lmask;               /* local copy of cantmask */
2236         klwp_t *lwp = ttolwp(curthread);

2238         /*
2239          * Mask out all signals except SIGHUP, SIGINT, SIGQUIT
2240          *    and SIGTERM. (Preserving the existing masks).
2241          *    This function supports the -intr nfs and ufs mount option.
2242          */

2244         /*
2245          * don't do kernel threads
2246          */
2247         if (lwp == NULL)
2248                 return;

2250         /*
2251          * get access to signal mask
2252          */
2253         p = ttoproc(curthread);
2254         owned = mutex_owned(&p->p_lock);        /* this is filthy */
2255         if (!owned)
2256                 mutex_enter(&p->p_lock);

2258         /*
2259          * remember the current mask
2260          */
2261         schedctl_finish_sigblock(curthread);
2262         *smask = curthread->t_hold;

2264         /*
2265          * mask out all signals
2266          */
2267         sigfillset(&curthread->t_hold);

2269         /*
2270          * Unmask the non-maskable signals (e.g., KILL), as long as
2271          * they aren't already masked (which could happen at exit).
2272          * The first sigdiffset sets lmask to (cantmask & ~curhold).  The
2273          * second sets the current hold mask to (~0 & ~lmask), which reduces
2274          * to (~cantmask | curhold).
2275          */
2276         lmask = cantmask;
2277         sigdiffset(&lmask, smask);
2278         sigdiffset(&curthread->t_hold, &lmask);

2280         /*
2281          * Re-enable HUP, QUIT, and TERM iff they were originally enabled
2282          * Re-enable INT if it's originally enabled and the NFS mount option
2283          * nointr is not set.
2284          */
2285         if (!sigismember(smask, SIGHUP))
2286                 sigdelset(&curthread->t_hold, SIGHUP);
2287         if (!sigismember(smask, SIGINT) && intable)
```



```
2288                 sigdelset(&curthread->t_hold, SIGINT);
2289         if (!sigismember(smask, SIGQUIT))
2290                 sigdelset(&curthread->t_hold, SIGQUIT);
2291         if (!sigismember(smask, SIGTERM))
2292                 sigdelset(&curthread->t_hold, SIGTERM);

2294         /*
2295          * release access to signal mask
2296          */
2297         if (!owned)
2298                 mutex_exit(&p->p_lock);

2300         /*
2301          * Indicate that this lwp is not to be stopped.
2302          */
2303         lwp->lwp_nostop++;

2305 }
2307 /* ONC_PLUS EXTRACT END */

2307 void
2308 sigunintr(k_sigset_t *smask)
2309 {
2310         proc_t *p;
2311         int owned;
2312         klwp_t *lwp = ttolwp(curthread);

2314         /*
2315          * Reset previous mask (See sigintr() above)
2316          */
2317         if (lwp != NULL) {
2318                 lwp->lwp_nostop--;      /* restore lwp stoppability */
2319                 p = ttoproc(curthread);
2320                 owned = mutex_owned(&p->p_lock);        /* this is filthy */
2321                 if (!owned)
2322                         mutex_enter(&p->p_lock);
2323                 curthread->t_hold = *smask;
2324                 /* so unmasked signals will be seen */
2325                 curthread->t_sig_check = 1;
2326                 if (!owned)
2327                         mutex_exit(&p->p_lock);
2328         }
2329 }
______unchanged_portion_omitted_
```

```
**********************************************************
   30849 Thu Jul 11 01:30:07 2013
new/usr/src/uts/common/os/swapgeneric.c
onc plus-be-gone
**********************************************************
```

```
/*
 * 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
 */
/* ONC_PLUS EXTRACT START */
/*
 * Copyright (c) 1982, 2010, Oracle and/or its affiliates. All rights reserved.
 */
/* ONC_PLUS EXTRACT END */

/*
 * Configure root, swap and dump devices.
 */

#include <sys/types.h>
#include <sys/param.h>
#include <sys/sysmacros.h>
#include <sys/signal.h>
#include <sys/cred.h>
#include <sys/proc.h>
#include <sys/user.h>
#include <sys/conf.h>
#include <sys/buf.h>
#include <sys/systm.h>
#include <sys/vm.h>
#include <sys/reboot.h>
#include <sys/file.h>
#include <sys/vfs.h>
#include <sys/vnode.h>
#include <sys/errno.h>
#include <sys/kmem.h>
#include <sys/uio.h>
#include <sys/open.h>
#include <sys/mount.h>
#include <sys/kobj.h>
#include <sys/bootconf.h>
#include <sys/sysconf.h>
#include <sys/modctl.h>
#include <sys/autoconf.h>
#include <sys/debug.h>
#include <sys/fs/snode.h>
#include <fs/fs_subr.h>
#include <sys/socket.h>
#include <net/if.h>
```



```
#include <sys/mkdev.h>
#include <sys/cmn_err.h>
#include <sys/console.h>

#include <sys/conf.h>
#include <sys/ddi.h>
#include <sys/sunddi.h>
#include <sys/hwconf.h>
#include <sys/dc_ki.h>
#include <sys/promif.h>
#include <sys/bootprops.h>

/*
 * Local routines
 */
static int preload_module(struct sysparam *, void *);
static struct vfssw *getfstype(char *, char *, size_t);
static int getphysdev(char *, char *, size_t);
static int load_bootpath_drivers(char *bootpath);
static int load_boot_driver(char *drv);
static int load_boot_platform_modules(char *drv);
static dev_info_t *path_to_devinfo(char *path);
static boolean_t netboot_over_ib(char *bootpath);
static boolean_t netboot_over_iscsi(void);

/*
 * Module linkage information for the kernel.
 */
static struct modlmisc modlmisc = {
        &mod_miscops, "root and swap configuration"
};
_____unchanged_portion_omitted_

/* ONC_PLUS EXTRACT START */
/*
 * We want to load all the modules needed to mount the root filesystem,
 * so that when we start the ball rolling in 'getrootdev', every module
 * should already be in memory, just waiting to be init-ed.
 */

int
loadrootmodules(void)
{
        struct vfssw    *vsw;
        char            *this;
        char            *name;
        int             err;
/* ONC_PLUS EXTRACT END */
        int             i, proplen;
        extern char     *impl_module_list[];
        extern char     *platform_module_list[];

        /* Make sure that the PROM's devinfo tree has been created */
        ASSERT(ddi_root_node());

        BMDPRINTF(("loadrootmodules: fstype %s\n", rootfs.bo_fstype));
        BMDPRINTF(("loadrootmodules: name %s\n", rootfs.bo_name));
        BMDPRINTF(("loadrootmodules: flags 0x%x\n", rootfs.bo_flags));

        /*
         * zzz We need to honor what's in rootfs if it's not null.
         * non-null means use what's there.  This way we can
         * change rootfs with /etc/system AND with tunetool.
         */
        if (root_is_svm) {
                /* user replaced rootdev, record obp_bootpath */
```

```
358                 obp_bootpath[0] = '\0';
359                 (void) getphysdev("root", obp_bootpath, BO_MAXOBJNAME);
360                 BMDPRINTF(("loadrootmodules: obp_bootpath %s\n", obp_bootpath));
361         } else {
362                 /*
363                  * Get the root fstype and root device path from boot.
364                  */
365                 rootfs.bo_fstype[0] = '\0';
366                 rootfs.bo_name[0] = '\0';
367         }
368
369         /*
370          * This lookup will result in modloadonly-ing the root
371          * filesystem module - it gets _init-ed in rootconf()
372          */
373         if ((vsw = getfstype("root", rootfs.bo_fstype, BO_MAXFSNAME)) == NULL)
374                 return (ENXIO); /* in case we have no file system types */
375
376         (void) strcpy(rootfs.bo_fstype, vsw->vsw_name);
377
378         vfs_unrefvfssw(vsw);
379
380         /*
381          * Load the favored drivers of the implementation.
382          * e.g. 'sbus' and possibly 'zs' (even).
383          *
384          * Called whilst boot is still loaded (because boot does
385          * the i/o for us), and DDI services are unavailable.
386          */
387         BMDPRINTF(("loadrootmodules: impl_module_list\n"));
388         for (i = 0; (this = impl_module_list[i]) != NULL; i++) {
389                 if ((err = load_boot_driver(this)) != 0) {
390                         cmn_err(CE_WARN, "Cannot load drv/%s", this);
391                         return (err);
392                 }
393         }
394         /*
395          * Now load the platform modules (if any)
396          */
397         BMDPRINTF(("loadrootmodules: platform_module_list\n"));
398         for (i = 0; (this = platform_module_list[i]) != NULL; i++) {
399                 if ((err = load_boot_platform_modules(this)) != 0) {
400                         cmn_err(CE_WARN, "Cannot load drv/%s", this);
401                         return (err);
402                 }
403         }
404
405 loop:
406         (void) getphysdev("root", rootfs.bo_name, BO_MAXOBJNAME);
407         /*
408          * Given a physical pathname, load the correct set of driver
409          * modules into memory, including all possible parents.
410          *
411          * NB: The code sets the variable 'name' for error reporting.
412          */
413         err = 0;
414         BMDPRINTF(("loadrootmodules: rootfs %s\n", rootfs.bo_name));
415         if (root_is_svm == 0) {
416                 BMDPRINTF(("loadrootmodules: rootfs %s\n", rootfs.bo_name));
417                 name = rootfs.bo_name;
418                 err = load_bootpath_drivers(rootfs.bo_name);
419         }
420
421         /*
422          * Load driver modules in obp_bootpath, this is always
423          * required for mountroot to succeed. obp_bootpath is
```



```
424          * is set if rootdev is set via /etc/system, which is
425          * the case if booting of a SVM/VxVM mirror.
426          */
427         if ((err == 0) && obp_bootpath[0] != '\0') {
428                 BMDPRINTF(("loadrootmodules: obp_bootpath %s\n", obp_bootpath));
429                 name = obp_bootpath;
430                 err = load_bootpath_drivers(obp_bootpath);
431         }
432
433         if (err != 0) {
434                 cmn_err(CE_CONT, "Cannot load drivers for %s\n", name);
435                 goto out;
436         }
437
438         /*
439          * Check to see if the booter performed DHCP configuration
440          * ("bootp-response" boot property exists). If so, then before
441          * bootops disappears we need to save the value of this property
442          * such that the userland dhcpagent can adopt the DHCP management
443          * of our primary network interface.
444          */
445         proplen = BOP_GETPROPLEN(bootops, "bootp-response");
446         if (proplen > 0) {
447                 dhcack = kmem_zalloc(proplen, KM_SLEEP);
448                 if (BOP_GETPROP(bootops, "bootp-response", dhcack) == -1) {
449                         cmn_err(CE_WARN, "BOP_GETPROP of  "
450                             "\"bootp-response\" failed\n");
451                         kmem_free(dhcack, dhcacklen);
452                         dhcack = NULL;
453                         goto out;
454                 }
455                 dhcacklen = proplen;
456
457                 /*
458                  * Fetch the "netdev-path" boot property (if it exists), and
459                  * stash it for later use by sysinfo(SI_DHCP_CACHE, ...).
460                  */
461                 proplen = BOP_GETPROPLEN(bootops, "netdev-path");
462                 if (proplen > 0) {
463                         netdev_path = kmem_zalloc(proplen, KM_SLEEP);
464                         if (BOP_GETPROP(bootops, "netdev-path",
465                             (uchar_t *)netdev_path) == -1) {
466                                 cmn_err(CE_WARN, "BOP_GETPROP of  "
467                                     "\"netdev-path\" failed\n");
468                                 kmem_free(netdev_path, proplen);
469                                 goto out;
470                         }
471                 }
472         }
473
474         /*
475          * Preload (load-only, no init) all modules which
476          * were added to the /etc/system file with the
477          * FORCELOAD keyword.
478          */
479         BMDPRINTF(("loadrootmodules: preload_module\n"));
480         (void) mod_sysctl_type(MOD_FORCELOAD, preload_module, NULL);
481
482 /* ONC_PLUS EXTRACT START */
483         /*
484          * If we booted otw then load in the plumbing
485          * routine now while we still can. If we didn't
486          * boot otw then we will load strplumb in main().
487          *
488          * NFS is actually a set of modules, the core routines,
489          * a diskless helper module, rpcmod, and the tli interface.  Load
```

```
489          * them now while we still can.
490          *
491          * Because we glomb all versions of nfs into a single module
492          * we check based on the initial string "nfs".
493          *
494          * XXX: A better test for this is to see if device_type
495          * XXX: from the PROM is "network".
496          */

498         if (strncmp(rootfs.bo_fstype, "nfs", 3) == 0) {
499                 ++netboot;

501                 /*
502                  * Preload (load-only, no init) the dacf module. We cannot
503                  * init the module because one of its requisite modules is
504                  * dld whose _init function will call taskq_create(), which
505                  * will panic the system at this point.
506                  */
507                 if ((err = modloadonly("dacf", "net_dacf")) < 0)  {
508                         cmn_err(CE_CONT, "Cannot load dacf/net_dacf\n");
509                         goto out;
510                 }
511                 if ((err = modload("misc", "tlimod")) < 0)  {
512                         cmn_err(CE_CONT, "Cannot load misc/tlimod\n");
513                         goto out;
514                 }
515                 if ((err = modload("strmod", "rpcmod")) < 0)  {
516                         cmn_err(CE_CONT, "Cannot load strmod/rpcmod\n");
517                         goto out;
518                 }
519                 if ((err = modload("misc", "nfs_dlboot")) < 0)  {
520                         cmn_err(CE_CONT, "Cannot load misc/nfs_dlboot\n");
521                         goto out;
522                 }
523                 if ((err = modload("mac", "mac_ether")) < 0)  {
524                         cmn_err(CE_CONT, "Cannot load mac/mac_ether\n");
525                         goto out;
526                 }
527                 if ((err = modload("misc", "strplumb")) < 0)  {
528                         cmn_err(CE_CONT, "Cannot load misc/strplumb\n");
529                         goto out;
530                 }
531                 if ((err = strplumb_load()) < 0) {
532                         goto out;
533                 }
534         }
535         if (netboot_over_iscsi() == B_TRUE) {
536                 /* iscsi boot */
537                 if ((err = modloadonly("dacf", "net_dacf")) < 0) {
538                         cmn_err(CE_CONT, "Cannot load dacf/net_dacf\n");
539                         goto out;
540                 }
541                 if ((err = modload("misc", "tlimod")) < 0) {
542                         cmn_err(CE_CONT, "Cannot load misc/tlimod\n");
543                         goto out;
544                 }
545                 if ((err = modload("mac", "mac_ether")) < 0) {
546                         cmn_err(CE_CONT, "Cannot load mac/mac_ether\n");
547                         goto out;
548                 }
549                 if ((err = modloadonly("drv", "iscsi")) < 0) {
550                         cmn_err(CE_CONT, "Cannot load drv/iscsi\n");
551                         goto out;
552                 }
553                 if ((err = modloadonly("drv", "ssd")) < 0) {
554                         cmn_err(CE_CONT, "Cannot load drv/ssd\n");
```



```
555                         goto out;
556                 }
557                 if ((err = modloadonly("drv", "sd")) < 0) {
558                         cmn_err(CE_CONT, "Cannot load drv/sd\n");
559                         goto out;
560                 }
561                 if ((err = modload("misc", "strplumb")) < 0) {
562                         cmn_err(CE_CONT, "Cannot load misc/strplumb\n");
563                         goto out;
564                 }
565                 if ((err = strplumb_load()) < 0) {
566                         goto out;
567                 }
568         }
569         /*
570          * Preload modules needed for booting as a cluster.
571          */
572         err = clboot_loadrootmodules();

574 out:
575         if (err != 0 && (boothowto & RB_ASKNAME))
576                 goto loop;

578         return (err);
579 }
585 /* ONC_PLUS EXTRACT END */

581 static int
582 get_bootpath_prop(char *bootpath)
583 {
584         if (root_is_ramdisk) {
585                 if (BOP_GETPROP(bootops, "bootarchive", bootpath) == -1)
586                         return (-1);
587                 (void) strlcat(bootpath, ":a", BO_MAXOBJNAME);
588         } else {
589                 /*
590                  * Look for the 1275 compliant name 'bootpath' first,
591                  * but make certain it has a non-NULL value as well.
592                  */
593                 if ((BOP_GETPROP(bootops, "bootpath", bootpath) == -1) ||
594                     strlen(bootpath) == 0) {
595                         if (BOP_GETPROP(bootops,
596                             "boot-path", bootpath) == -1)
597                                 return (-1);
598                 }
599                 if (memcmp(bootpath, BP_ISCSI_DISK,
600                     strlen(BP_ISCSI_DISK)) == 0) {
601                         /* iscsi boot */
602                         get_iscsi_bootpath_vhci(bootpath);
603                 }
604         }
605         return (0);
606 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   46233 Thu Jul 11 01:30:08 2013
new/usr/src/uts/common/os/sysent.c
onc plus-be-gone
**********************************************************
```

```
/*
 * 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
 */

/* ONC_PLUS EXTRACT START */

/*
 * Copyright (c) 1988, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright 2012 Milan Jurik. All rights reserved.
 * Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
 */

/*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */

/* ONC_PLUS EXTRACT END */

#include <sys/param.h>
#include <sys/types.h>
#include <sys/systm.h>
#include <sys/systrace.h>
#include <sys/procfs.h>
#include <sys/mman.h>
#include <sys/int_types.h>
#include <c2/audit.h>
#include <sys/stat.h>
#include <sys/times.h>
#include <sys/statfs.h>
#include <sys/stropts.h>
#include <sys/statvfs.h>
#include <sys/utsname.h>
#include <sys/timex.h>
#include <sys/socket.h>
#include <sys/sendfile.h>

struct hrtsysa;
struct mmaplf32a;

/* ONC_PLUS EXTRACT START */
/*
 * This table is the switch used to transfer to the appropriate
 * routine for processing a system call.  Each row contains the
 * number of arguments expected, a switch that tells systrap()
 * in trap.c whether a setjmp() is not necessary, and a pointer
```



```
 * to the routine.
 */
/* ONC_PLUS EXTRACT END */

int     access(char *, int);
int     alarm(int);
int     auditsys(struct auditcalls *, rval_t *);
int64_t brandsys(int, uintptr_t, uintptr_t, uintptr_t, uintptr_t, uintptr_t,
    uintptr_t);
int     brk(caddr_t);
int     chdir(char *);
int     chmod(char *, int);
int     chown(char *, uid_t, gid_t);
int     chroot(char *);
int     cladm(int, int, void *);
int     close(int);
int     exece(const char *, const char **, const char **);
int     faccessat(int, char *, int, int);
int     fchmodat(int, char *, int, int);
int     fchownat(int, char *, uid_t, gid_t, int);
int     fcntl(int, int, intptr_t);
int64_t vfork();
int64_t forksys(int, int);
int     fstat(int, struct stat *);
int     fdsync(int, int);
int64_t getgid();
int     ucredsys(int, int, void *);
int64_t getpid();
int64_t getuid();
time_t  gtime();
int     getloadavg(int *, int);
int     rusagesys(int, void *, void *, void *, void *);
int     getpagesizes(int, size_t *, int);
int     gtty(int, intptr_t);
#if defined(__i386) || defined(__amd64)
int     hrtsys(struct hrtsysa *, rval_t *);
#endif /* __i386 || __amd64 */
int     ioctl(int, int, intptr_t);
int     kill();
int     labelsys(int, void *, void *, void *, void *, void *);
int     link(char *, char *);
int     linkat(int, char *, int, char *, int);
off32_t lseek32(int32_t, off32_t, int32_t);
off_t   lseek64(int, off_t, int);
int     lgrpsys(int, long, void *);
int     mmapobjsys(int, uint_t, mmapobj_result_t *, uint_t *, void *);
int     mknod(char *, mode_t, dev_t);
int     mknodat(int, char *, mode_t, dev_t);
int     mount(long *, rval_t *);
int     nice(int);
int     nullsys();
int     open(char *, int, int);
int     openat(int, char *, int, int);
int     pause();
long    pcsample(void *, long);
int     privsys(int, priv_op_t, priv_ptype_t, void *, size_t, int);
int     profil(unsigned short *, size_t, ulong_t, uint_t);
ssize_t pread(int, void *, size_t, off_t);
ssize_t pwrite(int, void *, size_t, off_t);
ssize_t read(int, void *, size_t);
int     rename(char *, char *);
int     renameat(int, char *, int, char *);
void    rexit(int);
int     semsys();
int     setgid(gid_t);
int     setpgrp(int, int, int);
```

```
122 int     setuid(uid_t);
123 uintptr_t       shmsys();
124 uint64_t        sidsys(int, int, int, int);
125 int     sigprocmask(int, sigset_t *, sigset_t *);
126 int     sigsuspend(sigset_t);
127 int     sigaltstack(struct sigaltstack *, struct sigaltstack *);
128 int     sigaction(int, struct sigaction *, struct sigaction *);
129 int     sigpending(int, sigset_t *);
130 int     sigresend(int, siginfo_t *, sigset_t *);
131 int     sigtimedwait(sigset_t *, siginfo_t *, timespec_t *);
132 int     getsetcontext(int, void *);
133 int     stat(char *, struct stat *);
134 int     fstatat(int, char *, struct stat *, int);
135 int     stime(time_t);
136 int     stty(int, intptr_t);
137 int     syssync();
138 int     sysacct(char *);
139 clock_t times(struct tms *);
140 long    ulimit(int, long);
141 int     getrlimit32(int, struct rlimit32 *);
142 int     setrlimit32(int, struct rlimit32 *);
143 int     umask(int);
144 int     umount2(char *, int);
145 int     unlink(char *);
146 int     unlinkat(int, char *, int);
147 int     utimesys(int, uintptr_t, uintptr_t, uintptr_t, uintptr_t);
148 int64_t utssys32(void *, int, int, void *);
149 int64_t utssys64(void *, long, int, void *);
150 int     uucopy(const void *, void *, size_t);
151 ssize_t uucopystr(const char *, char *, size_t);
152 ssize_t write(int, void *, size_t);
153 ssize_t readv(int, struct iovec *, int);
154 ssize_t writev(int, struct iovec *, int);
155 int     syslwp_park(int, uintptr_t, uintptr_t);
156 int     rmdir(char *);
157 int     mkdir(char *, int);
158 int     mkdirat(int, char *, int);
159 int     getdents32(int, void *, size_t);
160 int     statfs32(char *, struct statfs32 *, int32_t, int32_t);
161 int     fstatfs32(int32_t, struct statfs32 *, int32_t, int32_t);
162 int     sysfs(int, long, long);
163 int     getmsg(int, struct strbuf *, struct strbuf *, int *);
164 int     pollsys(pollfd_t *, nfds_t, timespec_t *, sigset_t *);
165 int     putmsg(int, struct strbuf *, struct strbuf *, int);
166 int     uadmin();
167 int     lstat(char *, struct stat *);
168 int     symlink(char *, char *);
169 int     symlinkat(char *, int, char *);
170 ssize_t readlink(char *, char *, size_t);
171 ssize_t readlinkat(int, char *, char *, size_t);
172 int     resolvepath(char *, char *, size_t);
173 int     setgroups(int, gid_t *);
174 int     getgroups(int, gid_t *);
175 int     fchdir(int);
176 int     fchown(int, uid_t, uid_t);
177 int     fchmod(int, int);
178 int     getcwd(char *, size_t);
179 int     statvfs(char *, struct statvfs *);
180 int     fstatvfs(int, struct statvfs *);
181 offset_t llseek32(int32_t, uint32_t, uint32_t, int);

183 #if (defined(__i386) && !defined(__amd64)) || defined(__i386_COMPAT)
184 int     sysi86(short, uintptr_t, uintptr_t, uintptr_t);
185 #endif

187 int     acl(const char *, int, int, void *);
```



```
188 int     facl(int, int, int, void *);
189 long    priocntlsys(int, procset_t *, int, caddr_t, caddr_t);
190 int     waitsys(idtype_t, id_t, siginfo_t *, int);
191 int     sigsendsys(procset_t *, int);
192 int     mincore(caddr_t, size_t, char *);
193 caddr_t smmap64(caddr_t, size_t, int, int, int, off_t);
194 caddr_t smmap32(caddr32_t, size32_t, int, int, int, off32_t);
195 int     smmaplf32(struct mmaplf32a *, rval_t *);
196 int     mprotect(caddr_t, size_t, int);
197 int     munmap(caddr_t, size_t);
198 int     uname(struct utsname *);
199 int     lchown(char *, uid_t, gid_t);
200 int     getpmsg(int, struct strbuf *, struct strbuf *, int *, int *);
201 int     putpmsg(int, struct strbuf *, struct strbuf *, int, int);
202 int     memcntl(caddr_t, size_t, int, caddr_t, int, int);
203 long    sysconfig(int);
204 int     adjtime(struct timeval *, struct timeval *);
205 long    systeminfo(int, char *, long);
206 int     setegid(gid_t);
207 int     seteuid(uid_t);

209 int     setreuid(uid_t, uid_t);
210 int     setregid(gid_t, gid_t);
211 int     install_utrap(utrap_entry_t type, utrap_handler_t, utrap_handler_t *);
212 #ifdef __sparc
213 int     sparc_utrap_install(utrap_entry_t type, utrap_handler_t,
214     utrap_handler_t, utrap_handler_t *, utrap_handler_t *);
215 #endif

217 int     syslwp_create(ucontext_t *, int, id_t *);
218 void    syslwp_exit();
219 int     syslwp_suspend(id_t);
220 int     syslwp_continue(id_t);
221 int     syslwp_private(int, int, uintptr_t);
222 int     lwp_detach(id_t);
223 int     lwp_info(timestruc_t *);
224 int     lwp_kill(id_t, int);
225 int     lwp_self();
226 int64_t lwp_sigmask(int, uint_t, uint_t, uint_t, uint_t);
227 int     yield();
228 int     lwp_wait(id_t, id_t *);
229 int     lwp_mutex_timedlock(lwp_mutex_t *, timespec_t *, uintptr_t);
230 int     lwp_mutex_wakeup(lwp_mutex_t *, int);
231 int     lwp_mutex_unlock(lwp_mutex_t *);
232 int     lwp_mutex_trylock(lwp_mutex_t *, uintptr_t);
233 int     lwp_mutex_register(lwp_mutex_t *, caddr_t);
234 int     lwp_rwlock_sys(int, lwp_rwlock_t *, timespec_t *);
235 int     lwp_sema_post(lwp_sema_t *);
236 int     lwp_sema_timedwait(lwp_sema_t *, timespec_t *, int);
237 int     lwp_sema_trywait(lwp_sema_t *);
238 int     lwp_cond_wait(lwp_cond_t *, lwp_mutex_t *, timespec_t *, int);
239 int     lwp_cond_signal(lwp_cond_t *);
240 int     lwp_cond_broadcast(lwp_cond_t *);
241 caddr_t schedctl();

243 long    pathconf(char *, int);
244 long    fpathconf(int, int);
245 int     processor_bind(idtype_t, id_t, processorid_t, processorid_t *);
246 int     processor_info(processorid_t, processor_info_t *);
247 int     p_online(processorid_t, int);

249 /*
250  *      POSIX .4 system calls *
251  */
252 int     clock_gettime(clockid_t, timespec_t *);
253 int     clock_settime(clockid_t, timespec_t *);
```

```
254 int     clock_getres(clockid_t, timespec_t *);
255 int     timer_create(clockid_t, struct sigevent *, timer_t *);
256 int     timer_delete(timer_t);
257 int     timer_settime(timer_t, int, itimerspec_t *, itimerspec_t *);
258 int     timer_gettime(timer_t, itimerspec_t *);
259 int     timer_getoverrun(timer_t);
260 int     nanosleep(timespec_t *, timespec_t *);
261 int     sigqueue(pid_t, int, void *, int, int);
262 int     signotify(int, siginfo_t *, signotify_id_t *);

264 int     getdents64(int, void *, size_t);
265 int     stat64(char *, struct stat64 *);
266 int     lstat64(char *, struct stat64 *);
267 int     fstatat64(int, char *, struct stat64 *, int);
268 int     fstat64(int, struct stat64 *);
269 int     statvfs64(char *, struct statvfs64 *);
270 int     fstatvfs64(int, struct statvfs64 *);
271 int     setrlimit64(int, struct rlimit64 *);
272 int     getrlimit64(int, struct rlimit64 *);
273 int     pread64(int, void *, size32_t, uint32_t, uint32_t);
274 int     pwrite64(int, void *, size32_t, uint32_t, uint32_t);
275 int     open64(char *, int, int);
276 int     openat64(int, char *, int, int);

278 /*
279  * NTP syscalls
280  */

282 int ntp_gettime(struct ntptimeval *);
283 int ntp_adjtime(struct timex *);

285 /*
286  *      ++++++++++++++++++++++++
287  *      ++  SunOS4.1 Buyback  ++
288  *      ++++++++++++++++++++++++
289  *
290  *      fchroot, vhangup, gettimeofday
291  */

293 int     fchroot(int);
294 int     vhangup();
295 int     gettimeofday(struct timeval *);
296 int     getitimer(uint_t, struct itimerval *);
297 int     setitimer(uint_t, struct itimerval *, struct itimerval *);

299 int     corectl(int, uintptr_t, uintptr_t, uintptr_t);
300 int     modctl(int, uintptr_t, uintptr_t, uintptr_t, uintptr_t, uintptr_t);
301 int64_t loadable_syscall();
302 int64_t indir();

304 long    tasksys(int, projid_t, uint_t, void *, size_t);
305 long    rctlsys(int, char *, void *, void *, size_t, int);

307 long    zone();

309 int     so_socket(int, int, int, char *, int);
310 int     so_socketpair(int[2]);
311 int     bind(int, struct sockaddr *, socklen_t, int);
312 int     listen(int, int, int);
313 int     accept(int, struct sockaddr *, socklen_t *, int, int);
314 int     connect(int, struct sockaddr *, socklen_t, int);
315 int     shutdown(int, int, int);
316 ssize_t recv(int, void *, size_t, int);
317 ssize_t recvfrom(int, void *, size_t, int, struct sockaddr *, socklen_t *);
318 ssize_t recvmsg(int, struct nmsghdr *, int);
319 ssize_t send(int, void *, size_t, int);
```



```
320 ssize_t sendmsg(int, struct nmsghdr *, int);
321 ssize_t sendto(int, void *, size_t, int, struct sockaddr *, socklen_t);
322 int     getpeername(int, struct sockaddr *, socklen_t *, int);
323 int     getsockname(int, struct sockaddr *, socklen_t *, int);
324 int     getsockopt(int, int, int, void *, socklen_t *, int);
325 int     setsockopt(int, int, int, void *, socklen_t *, int);
326 int     sockconfig(int, void *, void *, void *, void *);
327 ssize_t sendfilev(int, int, const struct sendfilevec *, int, size_t *);

329 typedef int64_t (*llfcn_t)();   /* for casting one-word returns */

331 /*
332  * Sysent initialization macros.
333  *      These take the name string of the system call even though that isn't
334  *      currently used in the sysent entry.  This might be useful someday.
335  *
336  * Initialization macro for system calls which take their args in the C style.
337  * These system calls return the longlong_t return value and must call
338  * set_errno() to return an error.  For SPARC, narg must be at most six.
339  * For more args, use the SYSENT_AP() routine.
340  *
341  * We are able to return two distinct values to userland via the rval_t.
342  * At this time, that corresponds to one 64-bit quantity, or two 32-bit
343  * quantities.  The kernel does not currently need to return two 64-bit
344  * values, or one 128 bit value(!), but we may do one day, so the calling
345  * sequence between userland and the kernel should permit it.
346  *
347  * The interpretation of rval_t is provided by the sy_flags field
348  * which is used to determine how to arrange the results in registers
349  * (or on the stack) for return userland.
350  */
351         /* returns a 64-bit quantity for both ABIs */
352 #define SYSENT_C(name, call, narg)      \
353         { (narg), SE_64RVAL, NULL, NULL, (llfcn_t)(call) }

355         /* returns one 32-bit value for both ABIs: r_val1 */
356 #define SYSENT_CI(name, call, narg)     \
357         { (narg), SE_32RVAL1, NULL, NULL, (llfcn_t)(call) }

359         /* returns 2 32-bit values: r_val1 & r_val2 */
360 #define SYSENT_2CI(name, call, narg)    \
361         { (narg), SE_32RVAL1|SE_32RVAL2, NULL, NULL, (llfcn_t)(call) }

363 /*
364  * Initialization macro for system calls which take their args in the standard
365  * Unix style of a pointer to the arg structure and a pointer to the rval_t.
366  *
367  * Deprecated wherever possible (slower on some architectures, and trickier
368  * to maintain two flavours).
369  */
370 #define SYSENT_AP(name, call, narg)     \
371         { (narg), SE_64RVAL, (call), NULL, syscall_ap }

373 /*
374  * Conditional constructors to build the tables without #ifdef clutter
375  */
376 #if defined(_LP64)
377 #define IF_LP64(true, false)    true
378 #else
379 #define IF_LP64(true, false)    false
380 #endif

382 #if defined(__sparc)
383 #define IF_sparc(true, false)   true
384 #else
385 #define IF_sparc(true, false)   false
```

```
386 #endif

388 #if defined(__i386) && !defined(__amd64)
389 #define IF_i386(true, false)    true
390 #else
391 #define IF_i386(true, false)    false
392 #endif

394 #if defined(__i386) || defined(__amd64)
395 #define IF_x86(true, false)     true
396 #else
397 #define IF_x86(true, false)     false
398 #endif

400 #if (defined(__i386) && !defined(__amd64)) || defined(__i386_COMPAT)
401 #define IF_386_ABI(true, false) true
402 #else
403 #define IF_386_ABI(true, false) false
404 #endif

406 /*
407  * Define system calls that return a native 'long' quantity i.e. a 32-bit
408  * or 64-bit integer - depending on how the kernel is itself compiled
409  * e.g. read(2) returns 'ssize_t' in the kernel and in userland.
410  */
411 #define SYSENT_CL(name, call, narg)     \
412         IF_LP64(SYSENT_C(name, call, narg), SYSENT_CI(name, call, narg))

414 /*
415  * Initialization macro for loadable native system calls.
416  */
417 #define SYSENT_LOADABLE()       \
418         { 0, SE_LOADABLE, (int (*)())nosys, NULL, loadable_syscall }
425 /* ONC_PLUS EXTRACT END */

420 /*
421  * Initialization macro for loadable 32-bit compatibility system calls.
422  */
423 #define SYSENT_LOADABLE32()     SYSENT_LOADABLE()

425 #define SYSENT_NOSYS()          SYSENT_C("nosys", nosys, 0)

427 struct sysent nosys_ent = SYSENT_NOSYS();

436 /* ONC_PLUS EXTRACT START */
429 /*
430  * Native sysent table.
431  */
432 struct sysent sysent[NSYSCALL] =
433 {
442 /* ONC_PLUS EXTRACT END */
434         /*  0 */ IF_LP64(
435                         SYSENT_NOSYS(),
436                         SYSENT_C("indir",       indir,          1)),
437         /*  1 */ SYSENT_CI("exit",              rexit,          1),
438         /*  2 */ SYSENT_LOADABLE(),                         /* (was forkall) */
439         /*  3 */ SYSENT_CL("read",              read,           3),
440         /*  4 */ SYSENT_CL("write",             write,          3),
441         /*  5 */ SYSENT_CI("open",              open,           3),
442         /*  6 */ SYSENT_CI("close",             close,          1),
443         /*  7 */ SYSENT_CI("linkat",            linkat,         5),
444         /*  8 */ SYSENT_LOADABLE(),                         /* (was creat) */
445         /*  9 */ SYSENT_CI("link",              link,           2),
446         /* 10 */ SYSENT_CI("unlink",            unlink,         1),
447         /* 11 */ SYSENT_CI("symlinkat",         symlinkat,      3),
448         /* 12 */ SYSENT_CI("chdir",             chdir,          1),
```



```
449         /* 13 */ SYSENT_CL("time",              gtime,          0),
450         /* 14 */ SYSENT_CI("mknod",             mknod,          3),
451         /* 15 */ SYSENT_CI("chmod",             chmod,          2),
452         /* 16 */ SYSENT_CI("chown",             chown,          3),
453         /* 17 */ SYSENT_CI("brk",               brk,            1),
454         /* 18 */ SYSENT_CI("stat",              stat,           2),
455         /* 19 */ IF_LP64(
456                         SYSENT_CL("lseek",      lseek64,        3),
457                         SYSENT_CL("lseek",      lseek32,        3)),
458         /* 20 */ SYSENT_2CI("getpid",           getpid,         0),
459         /* 21 */ SYSENT_AP("mount",             mount,          8),
460         /* 22 */ SYSENT_CL("readlinkat",        readlinkat,     4),
461         /* 23 */ SYSENT_CI("setuid",            setuid,         1),
462         /* 24 */ SYSENT_2CI("getuid",           getuid,         0),
463         /* 25 */ SYSENT_CI("stime",             stime,          1),
464         /* 26 */ SYSENT_CL("pcsample",          pcsample,       2),
465         /* 27 */ SYSENT_CI("alarm",             alarm,          1),
466         /* 28 */ SYSENT_CI("fstat",             fstat,          2),
467         /* 29 */ SYSENT_CI("pause",             pause,          0),
468         /* 30 */ SYSENT_LOADABLE(),                        /* (was utime) */
469         /* 31 */ SYSENT_CI("stty",              stty,           2),
470         /* 32 */ SYSENT_CI("gtty",              gtty,           2),
471         /* 33 */ SYSENT_CI("access",            access,         2),
472         /* 34 */ SYSENT_CI("nice",              nice,           1),
473         /* 35 */ IF_LP64(
474                         SYSENT_NOSYS(),
475                         SYSENT_CI("statfs",     statfs32,       4)),
476         /* 36 */ SYSENT_CI("sync",              syssync,        0),
477         /* 37 */ SYSENT_CI("kill",              kill,           2),
478         /* 38 */ IF_LP64(
479                         SYSENT_NOSYS(),
480                         SYSENT_CI("fstatfs",    fstatfs32,      4)),
481         /* 39 */ SYSENT_CI("setpgrp",           setpgrp,        3),
482         /* 40 */ SYSENT_CI("uucopystr",         uucopystr,      3),
483         /* 41 */ SYSENT_LOADABLE(),                        /* (was dup) */
484         /* 42 */ SYSENT_LOADABLE(),                        /* pipe */
485         /* 43 */ SYSENT_CL("times",             times,          1),
486         /* 44 */ SYSENT_CI("prof",              profil,         4),
487         /* 45 */ SYSENT_CI("faccessat",         faccessat,      4),
488         /* 46 */ SYSENT_CI("setgid",            setgid,         1),
489         /* 47 */ SYSENT_2CI("getgid",           getgid,         0),
490         /* 48 */ SYSENT_CI("mknodat",           mknodat,        4),
491         /* 49 */ SYSENT_LOADABLE(),                        /* msgsys */
492         /* 50 */ IF_x86(
493                         SYSENT_CI("sysi86",     sysi86,         4),
494                         SYSENT_LOADABLE()),             /* (was sys3b) */
495         /* 51 */ SYSENT_LOADABLE(),                        /* sysacct */
496         /* 52 */ SYSENT_LOADABLE(),                        /* shmsys */
497         /* 53 */ SYSENT_LOADABLE(),                        /* semsys */
498         /* 54 */ SYSENT_CI("ioctl",             ioctl,          3),
499         /* 55 */ SYSENT_CI("uadmin",            uadmin,         3),
500         /* 56 */ SYSENT_CI("fchownat",          fchownat,       5),
501         /* 57 */ IF_LP64(
502                         SYSENT_2CI("utssys",    utssys64,       4),
503                         SYSENT_2CI("utssys",    utssys32,       4)),
504         /* 58 */ SYSENT_CI("fdsync",            fdsync,         2),
505         /* 59 */ SYSENT_CI("exece",             exece,          3),
506         /* 60 */ SYSENT_CI("umask",             umask,          1),
507         /* 61 */ SYSENT_CI("chroot",            chroot,         1),
508         /* 62 */ SYSENT_CI("fcntl",             fcntl,          3),
509         /* 63 */ SYSENT_CI("ulimit",            ulimit,         2),
510         /* 64 */ SYSENT_CI("renameat",          renameat,       4),
511         /* 65 */ SYSENT_CI("unlinkat",          unlinkat,       3),
512         /* 66 */ SYSENT_CI("fstatat",           fstatat,        4),
513         /* 67 */ IF_LP64(
514                         SYSENT_NOSYS(),
```

```
515                         SYSENT_CI("fstatat64",  fstatat64,      4)),
516         /* 68 */ SYSENT_CI("openat",           openat,         4),
517         /* 69 */ IF_LP64(
518                         SYSENT_NOSYS(),
519                         SYSENT_CI("openat64",   openat64,       4)),
520         /* 70 */ SYSENT_CI("tasksys",          tasksys,        5),
521         /* 71 */ SYSENT_LOADABLE(),            /* acctctl */
522         /* 72 */ SYSENT_LOADABLE(),            /* exacct */
523         /* 73 */ SYSENT_CI("getpagesizes",     getpagesizes,   3),
524         /* 74 */ SYSENT_CI("rctlsys",          rctlsys,        6),
525         /* 75 */ SYSENT_2CI("sidsys",          sidsys,         4),
526         /* 76 */ SYSENT_LOADABLE(),                        /* (was fsat) */
527         /* 77 */ SYSENT_CI("lwp_park",         syslwp_park,    3),
528         /* 78 */ SYSENT_CL("sendfilev",        sendfilev,      5),
529         /* 79 */ SYSENT_CI("rmdir",            rmdir,          1),
530         /* 80 */ SYSENT_CI("mkdir",            mkdir,          2),
531         /* 81 */ IF_LP64(
532                         SYSENT_CI("getdents",   getdents64,     3),
533                         SYSENT_CI("getdents",   getdents32,     3)),
534         /* 82 */ SYSENT_CI("privsys",          privsys,        6),
535         /* 83 */ SYSENT_CI("ucredsys",         ucredsys,       3),
536         /* 84 */ SYSENT_CI("sysfs",            sysfs,          3),
537         /* 85 */ SYSENT_CI("getmsg",           getmsg,         4),
538         /* 86 */ SYSENT_CI("putmsg",           putmsg,         4),
539         /* 87 */ SYSENT_LOADABLE(),                        /* (was poll) */
540         /* 88 */ SYSENT_CI("lstat",            lstat,          2),
541         /* 89 */ SYSENT_CI("symlink",          symlink,        2),
542         /* 90 */ SYSENT_CL("readlink",         readlink,       3),
543         /* 91 */ SYSENT_CI("setgroups",        setgroups,      2),
544         /* 92 */ SYSENT_CI("getgroups",        getgroups,      2),
545         /* 93 */ SYSENT_CI("fchmod",           fchmod,         2),
546         /* 94 */ SYSENT_CI("fchown",           fchown,         3),
547         /* 95 */ SYSENT_CI("sigprocmask",      sigprocmask,    3),
548         /* 96 */ SYSENT_CI("sigsuspend",       sigsuspend,     1),
549         /* 97 */ SYSENT_CI("sigaltstack",      sigaltstack,    2),
550         /* 98 */ SYSENT_CI("sigaction",        sigaction,      3),
551         /* 99 */ SYSENT_CI("sigpending",       sigpending,     2),
552         /* 100 */ SYSENT_CI("getsetcontext",   getsetcontext,  2),
553         /* 101 */ SYSENT_CI("fchmodat",        fchmodat,       4),
554         /* 102 */ SYSENT_CI("mkdirat",         mkdirat,        3),
555         /* 103 */ SYSENT_CI("statvfs",         statvfs,        2),
556         /* 104 */ SYSENT_CI("fstatvfs",        fstatvfs,       2),
557         /* 105 */ SYSENT_CI("getloadavg",      getloadavg,     2),
567 /* ONC_PLUS EXTRACT START */
558         /* 106 */ SYSENT_LOADABLE(),           /* nfssys */
569 /* ONC_PLUS EXTRACT END */
559         /* 107 */ SYSENT_CI("waitsys",         waitsys,        4),
560         /* 108 */ SYSENT_CI("sigsendset",      sigsendsys,     2),
561         /* 109 */ IF_x86(
562                         SYSENT_AP("hrtsys",     hrtsys,         5),
563                         SYSENT_LOADABLE()),
564         /* 110 */ SYSENT_CI("utimesys",        utimesys,       5),
565         /* 111 */ SYSENT_CI("sigresend",       sigresend,      3),
566         /* 112 */ SYSENT_CL("priocntlsys",     priocntlsys,    5),
567         /* 113 */ SYSENT_CL("pathconf",        pathconf,       2),
568         /* 114 */ SYSENT_CI("mincore",         mincore,        3),
569         /* 115 */ IF_LP64(
570                         SYSENT_CL("mmap",       smmap64,        6),
571                         SYSENT_CL("mmap",       smmap32,        6)),
572         /* 116 */ SYSENT_CI("mprotect",        mprotect,       3),
573         /* 117 */ SYSENT_CI("munmap",          munmap,         2),
574         /* 118 */ SYSENT_CL("fpathconf",       fpathconf,      2),
575         /* 119 */ SYSENT_2CI("vfork",          vfork,          0),
576         /* 120 */ SYSENT_CI("fchdir",          fchdir,         1),
577         /* 121 */ SYSENT_CL("readv",           readv,          3),
578         /* 122 */ SYSENT_CL("writev",          writev,         3),
```



```
579         /* 123 */ SYSENT_LOADABLE(),                   /* (was xstat) */
580         /* 124 */ SYSENT_LOADABLE(),                   /* (was lxstat) */
581         /* 125 */ SYSENT_LOADABLE(),                   /* (was fxstat) */
582         /* 126 */ SYSENT_LOADABLE(),                   /* (was xmknod) */
583         /* 127 */ SYSENT_CI("mmapobj",         mmapobjsys,     5),
584         /* 128 */ IF_LP64(
585                         SYSENT_CI("setrlimit",  setrlimit64,    2),
586                         SYSENT_CI("setrlimit",  setrlimit32,    2)),
587         /* 129 */ IF_LP64(
588                         SYSENT_CI("getrlimit",  getrlimit64,    2),
589                         SYSENT_CI("getrlimit",  getrlimit32,    2)),
590         /* 130 */ SYSENT_CI("lchown",          lchown,         3),
591         /* 131 */ SYSENT_CI("memcntl",         memcntl,        6),
592         /* 132 */ SYSENT_CI("getpmsg",         getpmsg,        5),
593         /* 133 */ SYSENT_CI("putpmsg",         putpmsg,        5),
594         /* 134 */ SYSENT_CI("rename",          rename,         2),
595         /* 135 */ SYSENT_CI("uname",           uname,          1),
596         /* 136 */ SYSENT_CI("setegid",         setegid,        1),
597         /* 137 */ SYSENT_CL("sysconfig",       sysconfig,      1),
598         /* 138 */ SYSENT_CI("adjtime",         adjtime,        2),
599         /* 139 */ SYSENT_CL("systeminfo",      systeminfo,     3),
600         /* 140 */ SYSENT_LOADABLE(),           /* sharefs */
601         /* 141 */ SYSENT_CI("seteuid",         seteuid,        1),
602         /* 142 */ SYSENT_2CI("forksys",        forksys,        2),
603         /* 143 */ SYSENT_LOADABLE(),                   /* (was fork1) */
604         /* 144 */ SYSENT_CI("sigtimedwait",    sigtimedwait,   3),
605         /* 145 */ SYSENT_CI("lwp_info",        lwp_info,       1),
606         /* 146 */ SYSENT_CI("yield",           yield,          0),
607         /* 147 */ SYSENT_LOADABLE(),               /* (was lwp_sema_wait) */
608         /* 148 */ SYSENT_CI("lwp_sema_post",   lwp_sema_post,  1),
609         /* 149 */ SYSENT_CI("lwp_sema_trywait", lwp_sema_trywait, 1),
610         /* 150 */ SYSENT_CI("lwp_detach",      lwp_detach,     1),
611         /* 151 */ SYSENT_CI("corectl",         corectl,        4),
612         /* 152 */ SYSENT_CI("modctl",          modctl,         6),
613         /* 153 */ SYSENT_CI("fchroot",         fchroot,        1),
614         /* 154 */ SYSENT_LOADABLE(),                   /* (was utimes) */
615         /* 155 */ SYSENT_CI("vhangup",         vhangup,        0),
616         /* 156 */ SYSENT_CI("gettimeofday",    gettimeofday,   1),
617         /* 157 */ SYSENT_CI("getitimer",       getitimer,      2),
618         /* 158 */ SYSENT_CI("setitimer",       setitimer,      3),
619         /* 159 */ SYSENT_CI("lwp_create",      syslwp_create,  3),
620         /* 160 */ SYSENT_CI("lwp_exit", (int (*)())syslwp_exit, 0),
621         /* 161 */ SYSENT_CI("lwp_suspend",     syslwp_suspend, 1),
622         /* 162 */ SYSENT_CI("lwp_continue",    syslwp_continue, 1),
623         /* 163 */ SYSENT_CI("lwp_kill",        lwp_kill,       2),
624         /* 164 */ SYSENT_CI("lwp_self",        lwp_self,       0),
625         /* 165 */ SYSENT_2CI("lwp_sigmask",    lwp_sigmask,    5),
626         /* 166 */ IF_x86(
627                         SYSENT_CI("lwp_private", syslwp_private, 3),
628                         SYSENT_NOSYS()),
629         /* 167 */ SYSENT_CI("lwp_wait",        lwp_wait,       2),
630         /* 168 */ SYSENT_CI("lwp_mutex_wakeup", lwp_mutex_wakeup,      2),
631         /* 169 */ SYSENT_LOADABLE(),               /* (was lwp_mutex_lock) */
632         /* 170 */ SYSENT_CI("lwp_cond_wait",   lwp_cond_wait,          4),
633         /* 171 */ SYSENT_CI("lwp_cond_signal", lwp_cond_signal,        1),
634         /* 172 */ SYSENT_CI("lwp_cond_broadcast", lwp_cond_broadcast,  1),
635         /* 173 */ SYSENT_CL("pread",           pread,          4),
636         /* 174 */ SYSENT_CL("pwrite ",         pwrite,         4),
637         /*
638          * The 64-bit C library maps llseek() to lseek(), so this
639          * is needed as a native syscall only on the 32-bit kernel.
640          */
641         /* 175 */ IF_LP64(
642                         SYSENT_NOSYS(),
643                         SYSENT_C("llseek",      llseek32,       4)),
644         /* 176 */ SYSENT_LOADABLE(),           /* inst_sync */
```

```
645         /* 177 */ SYSENT_CI("brandsys",         brandsys,       6),
646         /* 178 */ SYSENT_LOADABLE(),            /* kaio */
647         /* 179 */ SYSENT_LOADABLE(),            /* cpc */
648         /* 180 */ SYSENT_CI("lgrpsys",          lgrpsys,        3),
649         /* 181 */ SYSENT_CI("rusagesys",        rusagesys,      5),
650         /* 182 */ SYSENT_LOADABLE(),            /* portfs */
651         /* 183 */ SYSENT_CI("pollsys",          pollsys,        4),
652         /* 184 */ SYSENT_CI("labelsys",         labelsys,       5),
653         /* 185 */ SYSENT_CI("acl",              acl,            4),
654         /* 186 */ SYSENT_AP("auditsys",         auditsys,       6),
655         /* 187 */ SYSENT_CI("processor_bind",   processor_bind, 4),
656         /* 188 */ SYSENT_CI("processor_info",   processor_info, 2),
657         /* 189 */ SYSENT_CI("p_online",         p_online,       2),
658         /* 190 */ SYSENT_CI("sigqueue",         sigqueue,       5),
659         /* 191 */ SYSENT_CI("clock_gettime",    clock_gettime,  2),
660         /* 192 */ SYSENT_CI("clock_settime",    clock_settime,  2),
661         /* 193 */ SYSENT_CI("clock_getres",     clock_getres,   2),
662         /* 194 */ SYSENT_CI("timer_create",     timer_create,   3),
663         /* 195 */ SYSENT_CI("timer_delete",     timer_delete,   1),
664         /* 196 */ SYSENT_CI("timer_settime",    timer_settime,  4),
665         /* 197 */ SYSENT_CI("timer_gettime",    timer_gettime,  2),
666         /* 198 */ SYSENT_CI("timer_getoverrun", timer_getoverrun, 1),
667         /* 199 */ SYSENT_CI("nanosleep",        nanosleep,      2),
668         /* 200 */ SYSENT_CI("facl",             facl,           4),
669         /* 201 */ SYSENT_LOADABLE(),            /* door */
670         /* 202 */ SYSENT_CI("setreuid",         setreuid,       2),
671         /* 203 */ SYSENT_CI("setregid",         setregid,       2),
672         /* 204 */ SYSENT_CI("install_utrap",    install_utrap,  3),
673         /* 205 */ SYSENT_CI("signotify",        signotify,      3),
674         /* 206 */ SYSENT_CL("schedctl",         schedctl,       0),
675         /* 207 */ SYSENT_LOADABLE(),            /* pset */
676         /* 208 */ IF_sparc(
677                    SYSENT_CI("sparc_utrap_install", sparc_utrap_install, 5),
678                    SYSENT_NOSYS()),
679         /* 209 */ SYSENT_CI("resolvepath",      resolvepath,    3),
680         /* 210 */ SYSENT_CI("lwp_mutex_timedlock", lwp_mutex_timedlock, 3),
681         /* 211 */ SYSENT_CI("lwp_sema_timedwait", lwp_sema_timedwait,   3),
682         /* 212 */ SYSENT_CI("lwp_rwlock_sys",   lwp_rwlock_sys, 3),
683         /*
684          * Syscalls 213-225: 32-bit system call support for large files.
685          *
686          * (The 64-bit C library transparently maps these system calls
687          * back to their native versions, so almost all of them are only
688          * needed as native syscalls on the 32-bit kernel).
689          */
690         /* 213 */ IF_LP64(
691                       SYSENT_NOSYS(),
692                       SYSENT_CI("getdents64", getdents64,     3)),
693         /* 214 */ IF_LP64(
694                       SYSENT_NOSYS(),
695                       SYSENT_AP("smmaplf32",  smmaplf32,      7)),
696         /* 215 */ IF_LP64(
697                       SYSENT_NOSYS(),
698                       SYSENT_CI("stat64",     stat64,         2)),
699         /* 216 */ IF_LP64(
700                       SYSENT_NOSYS(),
701                       SYSENT_CI("lstat64",    lstat64,        2)),
702         /* 217 */ IF_LP64(
703                       SYSENT_NOSYS(),
704                       SYSENT_CI("fstat64",    fstat64,        2)),
705         /* 218 */ IF_LP64(
706                       SYSENT_NOSYS(),
707                       SYSENT_CI("statvfs64",  statvfs64,      2)),
708         /* 219 */ IF_LP64(
709                       SYSENT_NOSYS(),
710                       SYSENT_CI("fstatvfs64", fstatvfs64,     2)),
```



```
711         /* 220 */ IF_LP64(
712                       SYSENT_NOSYS(),
713                       SYSENT_CI("setrlimit64", setrlimit64,   2)),
714         /* 221 */ IF_LP64(
715                       SYSENT_NOSYS(),
716                       SYSENT_CI("getrlimit64", getrlimit64,   2)),
717         /* 222 */ IF_LP64(
718                       SYSENT_NOSYS(),
719                       SYSENT_CI("pread64",    pread64,        5)),
720         /* 223 */ IF_LP64(
721                       SYSENT_NOSYS(),
722                       SYSENT_CI("pwrite64",   pwrite64,       5)),
723         /* 224 */ SYSENT_LOADABLE(),                   /* (was creat64) */
724         /* 225 */ IF_LP64(
725                       SYSENT_NOSYS(),
726                       SYSENT_CI("open64",     open64,         3)),
727         /* 226 */ SYSENT_LOADABLE(),            /* rpcsys */
728         /* 227 */ SYSENT_CL("zone",             zone,           5),
729         /* 228 */ SYSENT_LOADABLE(),            /* autofssys */
730         /* 229 */ SYSENT_CI("getcwd",           getcwd,         2),
731         /* 230 */ SYSENT_CI("so_socket",        so_socket,      5),
732         /* 231 */ SYSENT_CI("so_socketpair",    so_socketpair,  1),
733         /* 232 */ SYSENT_CI("bind",             bind,           4),
734         /* 233 */ SYSENT_CI("listen",           listen,         3),
735         /* 234 */ SYSENT_CI("accept",           accept,         5),
736         /* 235 */ SYSENT_CI("connect",          connect,        4),
737         /* 236 */ SYSENT_CI("shutdown",         shutdown,       3),
738         /* 237 */ SYSENT_CL("recv",             recv,           4),
739         /* 238 */ SYSENT_CL("recvfrom",         recvfrom,       6),
740         /* 239 */ SYSENT_CL("recvmsg",          recvmsg,        3),
741         /* 240 */ SYSENT_CL("send",             send,           4),
742         /* 241 */ SYSENT_CL("sendmsg",          sendmsg,        3),
743         /* 242 */ SYSENT_CL("sendto",           sendto,         6),
744         /* 243 */ SYSENT_CI("getpeername",      getpeername,    4),
745         /* 244 */ SYSENT_CI("getsockname",      getsockname,    4),
746         /* 245 */ SYSENT_CI("getsockopt",       getsockopt,     6),
747         /* 246 */ SYSENT_CI("setsockopt",       setsockopt,     6),
748         /* 247 */ SYSENT_CI("sockconfig",       sockconfig,     5),
749         /* 248 */ SYSENT_CI("ntp_gettime",      ntp_gettime,    1),
750         /* 249 */ SYSENT_CI("ntp_adjtime",      ntp_adjtime,    1),
751         /* 250 */ SYSENT_CI("lwp_mutex_unlock", lwp_mutex_unlock,       1),
752         /* 251 */ SYSENT_CI("lwp_mutex_trylock", lwp_mutex_trylock,     2),
753         /* 252 */ SYSENT_CI("lwp_mutex_register", lwp_mutex_register,   2),
754         /* 253 */ SYSENT_CI("cladm",            cladm,          3),
755         /* 254 */ SYSENT_CI("uucopy",           uucopy,         3),
756         /* 255 */ SYSENT_CI("umount2",          umount2,        2)
768 /* ONC_PLUS EXTRACT START */
757 };
770 /* ONC_PLUS EXTRACT END */


760 #ifdef _SYSCALL32_IMPL

762 extern int ulimit32(int, int);
763 extern ssize_t read32(int32_t, caddr32_t, size32_t);
764 extern ssize_t write32(int32_t, caddr32_t, size32_t);
765 extern ssize_t pread32(int32_t, caddr32_t, size32_t, off32_t);
766 extern ssize_t pwrite32(int32_t, caddr32_t, size32_t, off32_t);
767 extern ssize_t readv32(int32_t, caddr32_t, int32_t);
768 extern ssize_t writev32(int32_t, caddr32_t, int32_t);
769 extern ssize_t readlink32(caddr32_t, caddr32_t, size32_t);
770 extern ssize_t readlinkat32(int, caddr32_t, caddr32_t, size32_t);
771 extern int open32(char *, int, int);
772 extern int openat32(int, char *, int, int);
773 extern int stat32(char *, struct stat32 *);
774 extern int fstatat32(int, char *, struct stat32 *, int);
```

```
775 extern int lstat32(char *, struct stat32 *);
776 extern int fstat32(int, struct stat32 *);
777 extern int fstatat64_32(int, char *, struct stat64_32 *, int);
778 extern int stat64_32(char *, struct stat64_32 *);
779 extern int lstat64_32(char *, struct stat64_32 *);
780 extern int fstat64_32(int, struct stat64_32 *);
781 extern int getmsg32(int, struct strbuf32 *, struct strbuf32 *, int32_t *);
782 extern int putmsg32(int, struct strbuf32 *, struct strbuf32 *, int32_t *);
783 extern int getpmsg32(int, struct strbuf32 *, struct strbuf32 *, int32_t *,
784     int32_t *);
785 extern int putpmsg32(int, struct strbuf32 *, struct strbuf32 *, int32_t,
786     int32_t);
787 extern int getsetcontext32(int, void *);
788 extern int statvfs32(char *, struct statvfs32 *);
789 extern int fstatvfs32(int, struct statvfs32 *);
790 extern int statvfs64_32(char *, struct statvfs64_32 *);
791 extern int fstatvfs64_32(int, struct statvfs64_32 *);
792 extern int sigaction32(int, struct sigaction32 *, struct sigaction32 *);
793 extern clock32_t times32(struct tms32 *);
794 extern int stime32(time32_t);
795 extern int getpagesizes32(int, size32_t *, int);
796 extern int sigaltstack32(struct sigaltstack32 *, struct sigaltstack32 *);
797 extern int sigqueue32(pid_t, int, caddr32_t, int, int);
798 extern offset_t llseek32(int32_t, uint32_t, uint32_t, int);
799 extern int waitsys32(idtype_t, id_t, siginfo_t *, int);

801 extern ssize_t recv32(int32_t, caddr32_t, size32_t, int32_t);
802 extern ssize_t recvfrom32(int32_t, caddr32_t, size32_t, int32_t, caddr32_t,
803     caddr32_t);
804 extern ssize_t send32(int32_t, caddr32_t, size32_t, int32_t);
805 extern ssize_t sendto32(int32_t, caddr32_t, size32_t, int32_t, caddr32_t,
806     socklen_t);

808 extern int privsys32(int, priv_op_t, priv_ptype_t, caddr32_t, size32_t, int);
809 extern int ucredsys32(int, int, caddr32_t);

824 /* ONC_PLUS EXTRACT START */
811 /*
812  * sysent table for ILP32 processes running on
813  * a LP64 kernel.
814  */
815 struct sysent sysent32[NSYSCALL] =
816 {
831 /* ONC_PLUS EXTRACT END */
817         /*  0 */ SYSENT_C("indir",              indir,          1),
818         /*  1 */ SYSENT_CI("exit",      (int (*)())rexit,       1),
819         /*  2 */ SYSENT_LOADABLE32(),                   /* (was forkall) */
820         /*  3 */ SYSENT_CI("read",              read32,         3),
821         /*  4 */ SYSENT_CI("write",             write32,        3),
822         /*  5 */ SYSENT_CI("open",              open32,         3),
823         /*  6 */ SYSENT_CI("close",             close,          1),
824         /*  7 */ SYSENT_CI("linkat",            linkat,         5),
825         /*  8 */ SYSENT_LOADABLE32(),                   /* (was creat32) */
826         /*  9 */ SYSENT_CI("link",              link,           2),
827         /* 10 */ SYSENT_CI("unlink",            unlink,         1),
828         /* 11 */ SYSENT_CI("symlinkat",         symlinkat,      3),
829         /* 12 */ SYSENT_CI("chdir",             chdir,          1),
830         /* 13 */ SYSENT_CI("time",              gtime,          0),
831         /* 14 */ SYSENT_CI("mknod",             mknod,          3),
832         /* 15 */ SYSENT_CI("chmod",             chmod,          2),
833         /* 16 */ SYSENT_CI("chown",             chown,          3),
834         /* 17 */ SYSENT_CI("brk",               brk,            1),
835         /* 18 */ SYSENT_CI("stat",              stat32,         2),
836         /* 19 */ SYSENT_CI("lseek",             lseek32,        3),
837         /* 20 */ SYSENT_2CI("getpid",           getpid,         0),
838         /* 21 */ SYSENT_AP("mount",             mount,          8),
```



```
839         /* 22 */ SYSENT_CI("readlinkat",        readlinkat32,   4),
840         /* 23 */ SYSENT_CI("setuid",            setuid,         1),
841         /* 24 */ SYSENT_2CI("getuid",           getuid,         0),
842         /* 25 */ SYSENT_CI("stime",             stime32,        1),
843         /* 26 */ SYSENT_CI("pcsample",          pcsample,       2),
844         /* 27 */ SYSENT_CI("alarm",             alarm,          1),
845         /* 28 */ SYSENT_CI("fstat",             fstat32,        2),
846         /* 29 */ SYSENT_CI("pause",             pause,          0),
847         /* 30 */ SYSENT_LOADABLE32(),                   /* (was utime) */
848         /* 31 */ SYSENT_CI("stty",              stty,           2),
849         /* 32 */ SYSENT_CI("gtty",              gtty,           2),
850         /* 33 */ SYSENT_CI("access",            access,         2),
851         /* 34 */ SYSENT_CI("nice",              nice,           1),
852         /* 35 */ SYSENT_CI("statfs",            statfs32,       4),
853         /* 36 */ SYSENT_CI("sync",              syssync,        0),
854         /* 37 */ SYSENT_CI("kill",              kill,           2),
855         /* 38 */ SYSENT_CI("fstatfs",           fstatfs32,      4),
856         /* 39 */ SYSENT_CI("setpgrp",           setpgrp,        3),
857         /* 40 */ SYSENT_CI("uucopystr",         uucopystr,      3),
858         /* 41 */ SYSENT_LOADABLE32(),                   /* (was dup) */
859         /* 42 */ SYSENT_LOADABLE32(),                   /* pipe */
860         /* 43 */ SYSENT_CI("times",             times32,        1),
861         /* 44 */ SYSENT_CI("prof",              profil,         4),
862         /* 45 */ SYSENT_CI("faccessat",         faccessat,      4),
863         /* 46 */ SYSENT_CI("setgid",            setgid,         1),
864         /* 47 */ SYSENT_2CI("getgid",           getgid,         0),
865         /* 48 */ SYSENT_CI("mknodat",           mknodat,        4),
866         /* 49 */ SYSENT_LOADABLE32(),                   /* msgsys */
867         /* 50 */ IF_386_ABI(
868                         SYSENT_CI("sysi86",     sysi86,         4),
869                         SYSENT_LOADABLE32()),           /* (was sys3b) */
870         /* 51 */ SYSENT_LOADABLE32(),                   /* sysacct */
871         /* 52 */ SYSENT_LOADABLE32(),                   /* shmsys */
872         /* 53 */ SYSENT_LOADABLE32(),                   /* semsys */
873         /* 54 */ SYSENT_CI("ioctl",             ioctl,          3),
874         /* 55 */ SYSENT_CI("uadmin",            uadmin,         3),
875         /* 56 */ SYSENT_CI("fchownat",          fchownat,       5),
876         /* 57 */ SYSENT_2CI("utssys",           utssys32,       4),
877         /* 58 */ SYSENT_CI("fdsync",            fdsync,         2),
878         /* 59 */ SYSENT_CI("exece",             exece,          3),
879         /* 60 */ SYSENT_CI("umask",             umask,          1),
880         /* 61 */ SYSENT_CI("chroot",            chroot,         1),
881         /* 62 */ SYSENT_CI("fcntl",             fcntl,          3),
882         /* 63 */ SYSENT_CI("ulimit",            ulimit32,       2),
883         /* 64 */ SYSENT_CI("renameat",          renameat,       4),
884         /* 65 */ SYSENT_CI("unlinkat",          unlinkat,       3),
885         /* 66 */ SYSENT_CI("fstatat",           fstatat32,      4),
886         /* 67 */ SYSENT_CI("fstatat64",         fstatat64_32,   4),
887         /* 68 */ SYSENT_CI("openat",            openat32,       4),
888         /* 69 */ SYSENT_CI("openat64",          openat64,       4),
889         /* 70 */ SYSENT_CI("tasksys",           tasksys,        5),
890         /* 71 */ SYSENT_LOADABLE32(),           /* acctctl */
891         /* 72 */ SYSENT_LOADABLE32(),           /* exacct */
892         /* 73 */ SYSENT_CI("getpagesizes",      getpagesizes32, 3),
893         /* 74 */ SYSENT_CI("rctlsys",           rctlsys,        6),
894         /* 75 */ SYSENT_2CI("sidsys",           sidsys,         4),
895         /* 76 */ SYSENT_LOADABLE32(),                   /* (was fsat) */
896         /* 77 */ SYSENT_CI("lwp_park",          syslwp_park,    3),
897         /* 78 */ SYSENT_CI("sendfilev",         sendfilev,      5),
898         /* 79 */ SYSENT_CI("rmdir",             rmdir,          1),
899         /* 80 */ SYSENT_CI("mkdir",             mkdir,          2),
900         /* 81 */ SYSENT_CI("getdents",          getdents32,     3),
901         /* 82 */ SYSENT_CI("privsys",           privsys32,      6),
902         /* 83 */ SYSENT_CI("ucredsys",          ucredsys32,     3),
903         /* 84 */ SYSENT_CI("sysfs",             sysfs,          3),
904         /* 85 */ SYSENT_CI("getmsg",            getmsg32,       4),
```

```
905         /* 86 */ SYSENT_CI("putmsg",            putmsg32,       4),
906         /* 87 */ SYSENT_LOADABLE32(),                   /* (was poll) */
907         /* 88 */ SYSENT_CI("lstat",             lstat32,        2),
908         /* 89 */ SYSENT_CI("symlink",           symlink,        2),
909         /* 90 */ SYSENT_CI("readlink",          readlink32,     3),
910         /* 91 */ SYSENT_CI("setgroups",         setgroups,      2),
911         /* 92 */ SYSENT_CI("getgroups",         getgroups,      2),
912         /* 93 */ SYSENT_CI("fchmod",            fchmod,         2),
913         /* 94 */ SYSENT_CI("fchown",            fchown,         3),
914         /* 95 */ SYSENT_CI("sigprocmask",       sigprocmask,    3),
915         /* 96 */ SYSENT_CI("sigsuspend",        sigsuspend,     1),
916         /* 97 */ SYSENT_CI("sigaltstack",       sigaltstack32,  2),
917         /* 98 */ SYSENT_CI("sigaction",         sigaction32,    3),
918         /* 99 */ SYSENT_CI("sigpending",        sigpending,     2),
919         /* 100 */ SYSENT_CI("getsetcontext",    getsetcontext32, 2),
920         /* 101 */ SYSENT_CI("fchmodat",         fchmodat,       4),
921         /* 102 */ SYSENT_CI("mkdirat",          mkdirat,        3),
922         /* 103 */ SYSENT_CI("statvfs",          statvfs32,      2),
923         /* 104 */ SYSENT_CI("fstatvfs",         fstatvfs32,     2),
924         /* 105 */ SYSENT_CI("getloadavg",       getloadavg,     2),
940 /* ONC_PLUS EXTRACT START */
925         /* 106 */ SYSENT_LOADABLE32(),          /* nfssys */
942 /* ONC_PLUS EXTRACT END */
926         /* 107 */ SYSENT_CI("waitsys",          waitsys32,      4),
927         /* 108 */ SYSENT_CI("sigsendset",       sigsendsys,     2),
928         /* 109 */ IF_x86(
929                       SYSENT_AP("hrtsys",       hrtsys,         5),
930                       SYSENT_LOADABLE32()),
931         /* 110 */ SYSENT_CI("utimesys",         utimesys,       5),
932         /* 111 */ SYSENT_CI("sigresend",        sigresend,      3),
933         /* 112 */ SYSENT_CI("priocntlsys",      priocntlsys,    5),
934         /* 113 */ SYSENT_CI("pathconf",         pathconf,       2),
935         /* 114 */ SYSENT_CI("mincore",          mincore,        3),
936         /* 115 */ SYSENT_CI("mmap",             smmap32,        6),
937         /* 116 */ SYSENT_CI("mprotect",         mprotect,       3),
938         /* 117 */ SYSENT_CI("munmap",           munmap,         2),
939         /* 118 */ SYSENT_CI("fpathconf",        fpathconf,      2),
940         /* 119 */ SYSENT_2CI("vfork",           vfork,          0),
941         /* 120 */ SYSENT_CI("fchdir",           fchdir,         1),
942         /* 121 */ SYSENT_CI("readv",            readv32,        3),
943         /* 122 */ SYSENT_CI("writev",           writev32,       3),
944         /* 123 */ SYSENT_LOADABLE32(),          /*      was xstat32     */
945         /* 124 */ SYSENT_LOADABLE32(),          /*      was lxstat32    */
946         /* 125 */ SYSENT_LOADABLE32(),          /*      was fxstat32    */
947         /* 126 */ SYSENT_LOADABLE32(),          /*      was xmknod      */
948         /* 127 */ SYSENT_CI("mmapobj",          mmapobjsys,     5),
949         /* 128 */ SYSENT_CI("setrlimit",        setrlimit32,    2),
950         /* 129 */ SYSENT_CI("getrlimit",        getrlimit32,    2),
951         /* 130 */ SYSENT_CI("lchown",           lchown,         3),
952         /* 131 */ SYSENT_CI("memcntl",          memcntl,        6),
953         /* 132 */ SYSENT_CI("getpmsg",          getpmsg32,      5),
954         /* 133 */ SYSENT_CI("putpmsg",          putpmsg32,      5),
955         /* 134 */ SYSENT_CI("rename",           rename,         2),
956         /* 135 */ SYSENT_CI("uname",            uname,          1),
957         /* 136 */ SYSENT_CI("setegid",          setegid,        1),
958         /* 137 */ SYSENT_CI("sysconfig",        sysconfig,      1),
959         /* 138 */ SYSENT_CI("adjtime",          adjtime,        2),
960         /* 139 */ SYSENT_CI("systeminfo",       systeminfo,     3),
961         /* 140 */ SYSENT_LOADABLE32(),          /* sharefs */
962         /* 141 */ SYSENT_CI("seteuid",          seteuid,        1),
963         /* 142 */ SYSENT_2CI("forksys",         forksys,        2),
964         /* 143 */ SYSENT_LOADABLE32(),                  /* (was fork1) */
965         /* 144 */ SYSENT_CI("sigtimedwait",     sigtimedwait,   3),
966         /* 145 */ SYSENT_CI("lwp_info",         lwp_info,       1),
967         /* 146 */ SYSENT_CI("yield",            yield,          0),
968         /* 147 */ SYSENT_LOADABLE32(),              /* (was lwp_sema_wait) */
```



```
969         /* 148 */ SYSENT_CI("lwp_sema_post",    lwp_sema_post,  1),
970         /* 149 */ SYSENT_CI("lwp_sema_trywait", lwp_sema_trywait, 1),
971         /* 150 */ SYSENT_CI("lwp_detach",       lwp_detach,     1),
972         /* 151 */ SYSENT_CI("corectl",          corectl,        4),
973         /* 152 */ SYSENT_CI("modctl",           modctl,         6),
974         /* 153 */ SYSENT_CI("fchroot",          fchroot,        1),
975         /* 154 */ SYSENT_LOADABLE32(),                  /* (was utimes) */
976         /* 155 */ SYSENT_CI("vhangup",          vhangup,        0),
977         /* 156 */ SYSENT_CI("gettimeofday",     gettimeofday,   1),
978         /* 157 */ SYSENT_CI("getitimer",        getitimer,      2),
979         /* 158 */ SYSENT_CI("setitimer",        setitimer,      3),
980         /* 159 */ SYSENT_CI("lwp_create",       syslwp_create,  3),
981         /* 160 */ SYSENT_CI("lwp_exit", (int (*))())syslwp_exit, 0),
982         /* 161 */ SYSENT_CI("lwp_suspend",      syslwp_suspend, 1),
983         /* 162 */ SYSENT_CI("lwp_continue",     syslwp_continue, 1),
984         /* 163 */ SYSENT_CI("lwp_kill",         lwp_kill,       2),
985         /* 164 */ SYSENT_CI("lwp_self",         lwp_self,       0),
986         /* 165 */ SYSENT_2CI("lwp_sigmask",     lwp_sigmask,    5),
987         /* 166 */ IF_x86(
988                       SYSENT_CI("lwp_private", syslwp_private, 3),
989                       SYSENT_NOSYS()),
990         /* 167 */ SYSENT_CI("lwp_wait",         lwp_wait,       2),
991         /* 168 */ SYSENT_CI("lwp_mutex_wakeup", lwp_mutex_wakeup,       2),
992         /* 169 */ SYSENT_LOADABLE32(),                  /* (was lwp_mutex_lock) */
993         /* 170 */ SYSENT_CI("lwp_cond_wait",    lwp_cond_wait,          4),
994         /* 171 */ SYSENT_CI("lwp_cond_signal",  lwp_cond_signal,        1),
995         /* 172 */ SYSENT_CI("lwp_cond_broadcast", lwp_cond_broadcast,   1),
996         /* 173 */ SYSENT_CI("pread",            pread32,                4),
997         /* 174 */ SYSENT_CI("pwrite",           pwrite32,               4),
998         /* 175 */ SYSENT_C("llseek",            llseek32,       4),
999         /* 176 */ SYSENT_LOADABLE32(),          /* inst_sync */
1000        /* 177 */ SYSENT_CI("brandsys",         brandsys,       6),
1001        /* 178 */ SYSENT_LOADABLE32(),          /* kaio */
1002        /* 179 */ SYSENT_LOADABLE32(),          /* cpc */
1003        /* 180 */ SYSENT_CI("lgrpsys",          lgrpsys,        3),
1004        /* 181 */ SYSENT_CI("rusagesys",        rusagesys,      5),
1005        /* 182 */ SYSENT_LOADABLE32(),          /* portfs */
1006        /* 183 */ SYSENT_CI("pollsys",          pollsys,        4),
1007        /* 184 */ SYSENT_CI("labelsys",         labelsys,       5),
1008        /* 185 */ SYSENT_CI("acl",              acl,            4),
1009        /* 186 */ SYSENT_AP("auditsys",         auditsys,       6),
1010        /* 187 */ SYSENT_CI("processor_bind",   processor_bind, 4),
1011        /* 188 */ SYSENT_CI("processor_info",   processor_info, 2),
1012        /* 189 */ SYSENT_CI("p_online",         p_online,       2),
1013        /* 190 */ SYSENT_CI("sigqueue",         sigqueue32,     5),
1014        /* 191 */ SYSENT_CI("clock_gettime",    clock_gettime,  2),
1015        /* 192 */ SYSENT_CI("clock_settime",    clock_settime,  2),
1016        /* 193 */ SYSENT_CI("clock_getres",     clock_getres,   2),
1017        /* 194 */ SYSENT_CI("timer_create",     timer_create,   3),
1018        /* 195 */ SYSENT_CI("timer_delete",     timer_delete,   1),
1019        /* 196 */ SYSENT_CI("timer_settime",    timer_settime,  4),
1020        /* 197 */ SYSENT_CI("timer_gettime",    timer_gettime,  2),
1021        /* 198 */ SYSENT_CI("timer_getoverrun", timer_getoverrun, 1),
1022        /* 199 */ SYSENT_CI("nanosleep",        nanosleep,      2),
1023        /* 200 */ SYSENT_CI("facl",             facl,           4),
1024        /* 201 */ SYSENT_LOADABLE32(),          /* door */
1025        /* 202 */ SYSENT_CI("setreuid",         setreuid,       2),
1026        /* 203 */ SYSENT_CI("setregid",         setregid,       2),
1027        /* 204 */ SYSENT_CI("install_utrap",    install_utrap,  3),
1028        /* 205 */ SYSENT_CI("signotify",        signotify,      3),
1029        /* 206 */ SYSENT_CI("schedctl",         schedctl,       0),
1030        /* 207 */ SYSENT_LOADABLE32(),          /* pset */
1031        /* 208 */ SYSENT_LOADABLE32(),
1032        /* 209 */ SYSENT_CI("resolvepath",      resolvepath,    3),
1033        /* 210 */ SYSENT_CI("lwp_mutex_timedlock", lwp_mutex_timedlock, 3),
1034        /* 211 */ SYSENT_CI("lwp_sema_timedwait", lwp_sema_timedwait,   3),
```

```
1035         /* 212 */ SYSENT_CI("lwp_rwlock_sys",   lwp_rwlock_sys, 3),
1036         /*
1037          * Syscalls 213-225: 32-bit system call support for large files.
1038          */
1039         /* 213 */ SYSENT_CI("getdents64",       getdents64,     3),
1040         /* 214 */ SYSENT_AP("smmaplf32",        smmaplf32,      7),
1041         /* 215 */ SYSENT_CI("stat64",           stat64_32,      2),
1042         /* 216 */ SYSENT_CI("lstat64",          lstat64_32,     2),
1043         /* 217 */ SYSENT_CI("fstat64",          fstat64_32,     2),
1044         /* 218 */ SYSENT_CI("statvfs64",        statvfs64_32,   2),
1045         /* 219 */ SYSENT_CI("fstatvfs64",       fstatvfs64_32,  2),
1046         /* 220 */ SYSENT_CI("setrlimit64",      setrlimit64,    2),
1047         /* 221 */ SYSENT_CI("getrlimit64",      getrlimit64,    2),
1048         /* 222 */ SYSENT_CI("pread64",          pread64,        5),
1049         /* 223 */ SYSENT_CI("pwrite64",         pwrite64,       5),
1050         /* 224 */ SYSENT_LOADABLE32(),                  /* (was creat64) */
1051         /* 225 */ SYSENT_CI("open64",           open64,         3),
1052         /* 226 */ SYSENT_LOADABLE32(),          /* rpcsys */
1053         /* 227 */ SYSENT_CI("zone",             zone,           6),
1054         /* 228 */ SYSENT_LOADABLE32(),          /* autofssys */
1055         /* 229 */ SYSENT_CI("getcwd",           getcwd,         2),
1056         /* 230 */ SYSENT_CI("so_socket",        so_socket,      5),
1057         /* 231 */ SYSENT_CI("so_socketpair",    so_socketpair,  1),
1058         /* 232 */ SYSENT_CI("bind",             bind,           4),
1059         /* 233 */ SYSENT_CI("listen",           listen,         3),
1060         /* 234 */ SYSENT_CI("accept",           accept,         5),
1061         /* 235 */ SYSENT_CI("connect",          connect,        4),
1062         /* 236 */ SYSENT_CI("shutdown",         shutdown,       3),
1063         /* 237 */ SYSENT_CI("recv",             recv32,         4),
1064         /* 238 */ SYSENT_CI("recvfrom",         recvfrom32,     6),
1065         /* 239 */ SYSENT_CI("recvmsg",          recvmsg,        3),
1066         /* 240 */ SYSENT_CI("send",             send32,         4),
1067         /* 241 */ SYSENT_CI("sendmsg",          sendmsg,        3),
1068         /* 242 */ SYSENT_CI("sendto",           sendto32,       6),
1069         /* 243 */ SYSENT_CI("getpeername",      getpeername,    4),
1070         /* 244 */ SYSENT_CI("getsockname",      getsockname,    4),
1071         /* 245 */ SYSENT_CI("getsockopt",       getsockopt,     6),
1072         /* 246 */ SYSENT_CI("setsockopt",       setsockopt,     6),
1073         /* 247 */ SYSENT_CI("sockconfig",       sockconfig,     5),
1074         /* 248 */ SYSENT_CI("ntp_gettime",      ntp_gettime,    1),
1075         /* 249 */ SYSENT_CI("ntp_adjtime",      ntp_adjtime,    1),
1076         /* 250 */ SYSENT_CI("lwp_mutex_unlock", lwp_mutex_unlock,       1),
1077         /* 251 */ SYSENT_CI("lwp_mutex_trylock", lwp_mutex_trylock,     2),
1078         /* 252 */ SYSENT_CI("lwp_mutex_register", lwp_mutex_register,   2),
1079         /* 253 */ SYSENT_CI("cladm",            cladm,          3),
1080         /* 254 */ SYSENT_CI("uucopy",           uucopy,         3),
1081         /* 255 */ SYSENT_CI("umount2",          umount2,        2)
1099 /* ONC_PLUS EXTRACT START */
1082 };
1101 /* ONC_PLUS EXTRACT END */
1083 #endif /* _SYSCALL32_IMPL */

1085 /*
1086  * Space allocated and initialized in init_syscallnames().
1087  */
1088 char **syscallnames;

1090 systrace_sysent_t *systrace_sysent;
1091 void (*systrace_probe)(dtrace_id_t, uintptr_t, uintptr_t,
1092     uintptr_t, uintptr_t, uintptr_t, uintptr_t);

1094 /*ARGSUSED*/
1095 void
1096 systrace_stub(dtrace_id_t id, uintptr_t arg0, uintptr_t arg1,
1097     uintptr_t arg2, uintptr_t arg3, uintptr_t arg4, uintptr_t arg5)
1098 {}
```



```
1100 /*ARGSUSED*/
1101 int64_t
1102 dtrace_systrace_syscall(uintptr_t arg0, uintptr_t arg1, uintptr_t arg2,
1103     uintptr_t arg3, uintptr_t arg4, uintptr_t arg5)
1104 {
1105         systrace_sysent_t *sy = &systrace_sysent[curthread->t_sysnum];
1106         dtrace_id_t id;
1107         int64_t rval;
1108         proc_t *p;

1110         if ((id = sy->stsy_entry) != DTRACE_IDNONE)
1111                 (*systrace_probe)(id, arg0, arg1, arg2, arg3, arg4, arg5);

1113         /*
1114          * We want to explicitly allow DTrace consumers to stop a process
1115          * before it actually executes the meat of the syscall.
1116          */
1117         p = ttoproc(curthread);
1118         mutex_enter(&p->p_lock);
1119         if (curthread->t_dtrace_stop && !curthread->t_lwp->lwp_nostop) {
1120                 curthread->t_dtrace_stop = 0;
1121                 stop(PR_REQUESTED, 0);
1122         }
1123         mutex_exit(&p->p_lock);

1125         rval = (*sy->stsy_underlying)(arg0, arg1, arg2, arg3, arg4, arg5);

1127         if (ttolwp(curthread)->lwp_errno != 0)
1128                 rval = -1;

1130         if ((id = sy->stsy_return) != DTRACE_IDNONE)
1131                 (*systrace_probe)(id, (uintptr_t)rval, (uintptr_t)rval,
1132                     (uintptr_t)((int64_t)rval >> 32), 0, 0, 0);

1134         return (rval);
1135 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2606 Thu Jul 11 01:30:09 2013
new/usr/src/uts/common/rpc/Makefile
first pass
**********************************************************
#
# 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
#

#
# ident "%Z%%M% %I%     %E% SMI"
#
# Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# uts/common/rpc/Makefile
#
# include global definitions
include ../../../Makefile.master

i386_HDRS=

sparc_HDRS= ib.h

COMMHDRS= \
auth.h          auth_des.h      auth_sys.h      auth_unix.h \
bootparam.h     clnt.h          clnt_soc.h      clnt_stat.h     des_crypt.h \
nettype.h       pmap_clnt.h     pmap_rmt.h \
raw.h           rpc.h           rpc_com.h       rpc_msg.h \
rpcb_clnt.h     rpcent.h        svc.h           svc_auth.h      svc_soc.h \
types.h         xdr.h           rpcsec_gss.h    svc_mt.h \
rpcsys.h        rpc_rdma.h

HDRS=                   \
        $(COMMHDRS)     \
        $($(MACH)_HDRS)

RPC_SRC=        pmap_prot.x     rpcb_prot.x

RPCSVC_SRC=     key_prot.x      rpc_sztypes.x

DERIVED_FILES=  key_prot.h      pmap_prot.h     rpcb_prot.h     rpc_sztypes.h

RPCHDRS=        $(HDRS) $(RPC_SRC) $(DERIVED_FILES)

RPCSVCHDRS=     $(RPCSVC_SRC)

RPCDIRS=        $(ROOT)/usr/include/rpc
```



```
RPCSVCDIRS=     $(ROOT)/usr/include/rpcsvc

ROOTHDRS= $(RPCHDRS:%=$(RPCDIRS)/%) $(RPCSVCHDRS:%=$(RPCSVCDIRS)/%)

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

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

# XXX: should really check the style of the derived files as well...
#       $(RPC_SRC:%.x=%.check) \
#       $(RPCSVC_SRC:%.x=%.check)
#
CHECKHDRS= $(HDRS:%.h=%.check)

.KEEP_STATE:

.PARALLEL: $(CHECKHDRS)

all: all_h

install_h: all_h $(RPCDIRS) $(RPCSVCDIRS) $(ROOTHDRS)

# all_h permits derived headers to be built here in the uts source area
# for the kernel to reference, without going so far as to install them.
#
all_h: $(DERIVED_FILES)

clean:
        $(RM) $(DERIVED_FILES)

$(RPCDIRS):
        $(INS.dir)

$(RPCSVCDIRS):
        $(INS.dir)

key_prot.h: key_prot.x
        $(RPCGEN) -C -h key_prot.x > $@

pmap_prot.h: pmap_prot.x
        $(RPCGEN) -h pmap_prot.x > $@

# EXPORT DELETE START
# Special target to clean up the source tree for export distribution
# Warning: This target changes the source tree
EXPORT_SRC:
        $(RM) Makefile+ sec_gss/rpcsec_gss_misc.c+
        $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(SED) -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
        < sec_gss/rpcsec_gss_misc.c > sec_gss/rpcsec_gss_misc.c+
        $(MV) sec_gss/rpcsec_gss_misc.c+ sec_gss/rpcsec_gss_misc.c
        $(CHMOD) 444 Makefile sec_gss/rpcsec_gss_misc.c

# EXPORT DELETE END

rpc_sztypes.h: rpc_sztypes.x
        $(RPCGEN) -C -h rpc_sztypes.x > $@

rpcb_prot.h: rpcb_prot.x
        $(RPCGEN) -h rpcb_prot.x > $@

check:  $(CHECKHDRS)
```

****
8168 Thu Jul 11 01:30:09 2013
new/usr/src/uts/common/rpc/sec_gss/rpcsec_gss_misc.c
first pass
****
_____unchanged_portion_omitted_

```
130 /*
131  * Generic routine to wrap data used by client and server sides.
132  */
133 bool_t
134 __rpc_gss_wrap_data(service, qop, context, seq_num, out_xdrs,
135                     xdr_func, xdr_ptr)
136         OM_uint32               qop;
137         rpc_gss_service_t       service;
138         gss_ctx_id_t            context;
139         uint_t                  seq_num;
140         XDR                     *out_xdrs;
141         bool_t                  (*xdr_func)();
142         caddr_t                 xdr_ptr;
143 {
144         OM_uint32               major, minor;
145         gss_buffer_desc         in_buf, out_buf;
146         XDR                     temp_xdrs;
147         char                    *temp_data;
148 /* EXPORT DELETE START */
148         bool_t                  conf_state;
150 /* EXPORT DELETE END */
149         bool_t                  ret = FALSE;
150         int                     size;

152         /*
153          * Create a temporary XDR/buffer to hold the data to be wrapped.
154          * We need an extra bit for the sequence number serialized first.
155          */
156         size = xdr_sizeof(xdr_func, xdr_ptr) + BYTES_PER_XDR_UNIT;
157         temp_data = kmem_alloc(size, KM_SLEEP);
158         out_buf.length = 0;

160         xdrmem_create(&temp_xdrs, temp_data, size, XDR_ENCODE);

162         /*
163          * serialize the sequence number into tmp memory
164          */
165         if (!xdr_u_int(&temp_xdrs, &seq_num))
166                 goto fail;

168         /*
169          * serialize the arguments into tmp memory
170          */
171         if (!(*xdr_func)(&temp_xdrs, xdr_ptr))
172                 goto fail;

174         /*
175          * Data to be wrapped goes in in_buf.  If privacy is used,
176          * out_buf will have wrapped data (in_buf will no longer be
177          * needed).  If integrity is used, out_buf will have checksum
178          * which will follow the data in in_buf.
179          */
180         in_buf.length = xdr_getpos(&temp_xdrs);
181         in_buf.value = (char *)temp_xdrs.x_base;

183         switch (service) {
184         case rpc_gss_svc_privacy:

188 /* EXPORT DELETE START */
```



```
186                 if ((major = kgss_seal(&minor, context, TRUE, qop, &in_buf,
187                                        &conf_state, &out_buf)) != GSS_S_COMPLETE) {
188                         RPCGSS_LOG1(1, "rpc_gss_wrap: kgss_seal failed."
189                                        "major = %x, minor = %x", major, minor);
190                         goto fail;
191                 }
192                 in_buf.length = 0;      /* in_buf not needed */
193                 if (!conf_state)
197 /* EXPORT DELETE END */
194                         goto fail;
199 /* EXPORT DELETE START */
195                 break;
201 /* EXPORT DELETE END */
196         case rpc_gss_svc_integrity:
197                 if ((major = kgss_sign(&minor, context, qop, &in_buf,
198                                        &out_buf)) != GSS_S_COMPLETE) {
199                         RPCGSS_LOG1(1, "rpc_gss_wrap: kgss_sign failed."
200                                        "major = %x, minor = %x", major, minor);
201                         goto fail;
202                 }
203                 break;
204         default:
205                 goto fail;
206         }

208         /*
209          * write out in_buf and out_buf as needed
210          */
211         if (in_buf.length != 0) {
212                 if (!__xdr_gss_buf(out_xdrs, &in_buf))
213                         goto fail;
214         }

216         if (!__xdr_gss_buf(out_xdrs, &out_buf))
217                 goto fail;
218         ret = TRUE;
219 fail:
220         kmem_free(temp_data, size);
221         if (out_buf.length != 0)
222                 (void) gss_release_buffer(&minor, &out_buf);
223         return (ret);
224 }

226 /*
227  * Generic routine to unwrap data used by client and server sides.
228  */
229 bool_t
230 __rpc_gss_unwrap_data(service, context, seq_num, qop_check, in_xdrs,
231                       xdr_func, xdr_ptr)
232         rpc_gss_service_t       service;
233         gss_ctx_id_t            context;
234         uint_t                  seq_num;
235         OM_uint32               qop_check;
236         XDR                     *in_xdrs;
237         bool_t                  (*xdr_func)();
238         caddr_t                 xdr_ptr;
239 {
240         gss_buffer_desc         in_buf, out_buf;
241         XDR                     temp_xdrs;
242         uint_t                  seq_num2;
243         bool_t                  conf = FALSE;
244         OM_uint32               major = GSS_S_COMPLETE, minor = 0;
245         int                     qop = 0;

247         in_buf.value = NULL;
248         out_buf.value = NULL;
```

```
250         /*
251          * Pull out wrapped data.  For privacy service, this is the
252          * encrypted data.  For integrity service, this is the data
253          * followed by a checksum.
254          */
255         if (!__xdr_gss_buf(in_xdrs, &in_buf)) {
256                 return (FALSE);
257         }

259         if (service == rpc_gss_svc_privacy) {
260                 major = GSS_S_FAILURE;
261 /* EXPORT DELETE START */
262                 major = kgss_unseal(&minor, context, &in_buf, &out_buf, &conf,
263                                     &qop);
270 /* EXPORT DELETE END */
263                 kmem_free(in_buf.value, in_buf.length);
264                 if (major != GSS_S_COMPLETE) {
265                         RPCGSS_LOG1(1, "rpc_gss_unwrap: kgss_unseal failed."
266                                 "major = %x, minor = %x", major, minor);
267                         return (FALSE);
268                 }
269                 /*
270                  * Keep the returned token (unencrypted data) in in_buf.
271                  */
272                 in_buf.length = out_buf.length;
273                 in_buf.value = out_buf.value;

275                 /*
276                  * If privacy was not used, or if QOP is not what we are
277                  * expecting, fail.
278                  */
279                 if (!conf || qop != qop_check)
280                         goto fail;

282         } else if (service == rpc_gss_svc_integrity) {
283                 if (!__xdr_gss_buf(in_xdrs, &out_buf)) {
284                         return (FALSE);
285                 }
286                 major = kgss_verify(&minor, context, &in_buf, &out_buf,
287                                     &qop);
288                 kmem_free(out_buf.value, out_buf.length);
289                 if (major != GSS_S_COMPLETE) {
290                         kmem_free(in_buf.value, in_buf.length);
291                         RPCGSS_LOG1(1, "rpc_gss_unwrap: kgss_verify failed."
292                                 "major = %x, minor = %x", major, minor);
293                         return (FALSE);
294                 }

296                 /*
297                  * If QOP is not what we are expecting, fail.
298                  */
299                 if (qop != qop_check)
300                         goto fail;
301         }

303         xdrmem_create(&temp_xdrs, in_buf.value, in_buf.length, XDR_DECODE);

305         /*
306          * The data consists of the sequence number followed by the
307          * arguments.  Make sure sequence number is what we are
308          * expecting (i.e., the value in the header).
309          */
310         if (!xdr_u_int(&temp_xdrs, &seq_num2))
311                 goto fail;
312         if (seq_num2 != seq_num)
```



```
313                 goto fail;

315         /*
316          * Deserialize the arguments into xdr_ptr, and release in_buf.
317          */
318         if (!(*xdr_func)(&temp_xdrs, xdr_ptr)) {
319                 goto fail;
320         }

322         if (service == rpc_gss_svc_privacy)
323                 (void) gss_release_buffer(&minor, &in_buf);
324         else
325                 kmem_free(in_buf.value, in_buf.length);
326         XDR_DESTROY(&temp_xdrs);
327         return (TRUE);
328 fail:
329         XDR_DESTROY(&temp_xdrs);
330         if (service == rpc_gss_svc_privacy)
331                 (void) gss_release_buffer(&minor, &in_buf);
332         else
333                 kmem_free(in_buf.value, in_buf.length);
334         return (FALSE);
335 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   22329 Thu Jul 11 01:30:10 2013
new/usr/src/uts/common/sys/Makefile
first pass
**********************************************************
#
# 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.
#

include $(SRC)/uts/Makefile.uts

FILEMODE=644

#
#       Note that the following headers are present in the kernel but
#       neither installed or shipped as part of the product:
#               cpuid_drv.h:            Private interface for cpuid consumers
#               unix_bb_info.h:         Private interface to kcov
#

i386_HDRS=                      \
        agp/agpamd64gart_io.h   \
        agp/agpdefs.h           \
        agp/agpgart_impl.h      \
        agp/agpmaster_io.h      \
        agp/agptarget_io.h      \
        agpgart.h               \
        asy.h                   \
        fd_debug.h              \
        fdc.h                   \
        fdmedia.h               \
        mouse.h                 \
        ucode.h

sparc_HDRS=                     \
        mouse.h                 \
        scsi/targets/ssddef.h   \
        $(MDESCHDRS)

# Generated headers
GENHDRS=                        \
        priv_const.h            \
        priv_names.h            \
        usb/usbdevs.h

CHKHDRS=                        \
```



```
        acpi_drv.h              \
        acct.h                  \
        acctctl.h               \
        acl.h                   \
        acl_impl.h              \
        aggr.h                  \
        aggr_impl.h             \
        aio.h                   \
        aio_impl.h              \
        aio_req.h               \
        aiocb.h                 \
        ascii.h                 \
        asynch.h                \
        atomic.h                \
        attr.h                  \
        audio.h                 \
        audioio.h               \
        autoconf.h              \
        auxv.h                  \
        auxv_386.h              \
        auxv_SPARC.h            \
        avl.h                   \
        avl_impl.h              \
        bitmap.h                \
        bitset.h                \
        bl.h                    \
        blkdev.h                \
        bofi.h                  \
        bofi_impl.h             \
        bpp_io.h                \
        bootstat.h              \
        brand.h                 \
        buf.h                   \
        bufmod.h                \
        bustypes.h              \
        byteorder.h             \
        callb.h                 \
        callo.h                 \
        cap_util.h              \
        cpucaps.h               \
        cpucaps_impl.h          \
        ccompile.h              \
        cdio.h                  \
        cladm.h                 \
        class.h                 \
        clconf.h                \
        clock_impl.h            \
        cmlb.h                  \
        cmn_err.h               \
        compress.h              \
        condvar.h               \
        condvar_impl.h          \
        conf.h                  \
        consdev.h               \
        console.h               \
        consplat.h              \
        vt.h                    \
        vtdaemon.h              \
        kd.h                    \
        contract.h              \
        contract_impl.h         \
        copyops.h               \
        core.h                  \
        corectl.h               \
        cpc_impl.h              \
        cpc_pcbe.h              \
```

```
128         cpr.h                   \
129         cpupart.h               \
130         cpuvar.h                \
131         crc32.h                 \
132         cred.h                  \
133         cred_impl.h             \
134         crtctl.h                \
135         cryptmod.h              \
136         csiioctl.h              \
137         ctf.h                   \
138         ctfs.h                  \
139         ctfs_impl.h             \
140         ctf_api.h               \
141         ctype.h                 \
142         cyclic.h                \
143         cyclic_impl.h           \
144         dacf.h                  \
145         dacf_impl.h             \
146         damap.h                 \
147         damap_impl.h            \
148         dc_ki.h                 \
149         ddi.h                   \
150         ddifm.h                 \
151         ddifm_impl.h            \
152         ddi_hp.h                \
153         ddi_hp_impl.h           \
154         ddi_intr.h              \
155         ddi_intr_impl.h         \
156         ddi_impldefs.h          \
157         ddi_implfuncs.h         \
158         ddi_obsolete.h          \
159         ddi_timer.h             \
160         ddidevmap.h             \
161         ddidmareq.h             \
162         ddimapreq.h             \
163         ddipropdefs.h           \
164         dditypes.h              \
165         debug.h                 \
166         des.h                   \
167         devctl.h                \
168         devcache.h              \
169         devcache_impl.h         \
170         devfm.h                 \
171         devid_cache.h           \
172         devinfo_impl.h          \
173         devops.h                \
174         devpolicy.h             \
175         devpoll.h               \
176         dirent.h                \
177         disp.h                  \
178         dkbad.h                 \
179         dkio.h                  \
180         dklabel.h               \
181         dl.h                    \
182         dlpi.h                  \
183         dld.h                   \
184         dld_impl.h              \
185         dld_ioc.h               \
186         dls.h                   \
187         dls_mgmt.h              \
188         dls_impl.h              \
189         dma_i8237A.h            \
190         dnlc.h                  \
191         door.h                  \
192         door_data.h             \
193         door_impl.h             \
```



```
194         dtrace.h                \
195         dtrace_impl.h           \
196         dumpadm.h               \
197         dumphdr.h               \
198         ecppsys.h               \
199         ecppio.h                \
200         ecppreg.h               \
201         ecppvar.h               \
202         efi_partition.h         \
203         elf.h                   \
204         elf_386.h               \
205         elf_SPARC.h             \
206         elf_notes.h             \
207         elf_amd64.h             \
208         elftypes.h              \
209         emul64.h                \
210         emul64cmd.h             \
211         emul64var.h             \
212         epm.h                   \
213         errno.h                 \
214         errorq.h                \
215         errorq_impl.h           \
216         esunddi.h               \
217         ethernet.h              \
218         euc.h                   \
219         eucioctl.h              \
220         exacct.h                \
221         exacct_catalog.h        \
222         exacct_impl.h           \
223         exec.h                  \
224         exechdr.h               \
225         extdirent.h             \
226         fault.h                 \
227         fasttrap.h              \
228         fasttrap_impl.h         \
229         fbio.h                  \
230         fbuf.h                  \
231         fcntl.h                 \
232         fct.h                   \
233         fct_defines.h           \
234         fctio.h                 \
235         fdbuffer.h              \
236         fdio.h                  \
237         feature_tests.h         \
238         fem.h                   \
239         file.h                  \
240         filio.h                 \
241         flock.h                 \
242         flock_impl.h            \
243         fork.h                  \
244         fss.h                   \
245         fsspriocntl.h           \
246         fsid.h                  \
247         fssnap.h                \
248         fssnap_if.h             \
249         fstyp.h                 \
250         ftrace.h                \
251         fx.h                    \
252         fxpriocntl.h            \
253         gfs.h                   \
254         gld.h                   \
255         gldpriv.h               \
256         group.h                 \
257         hdio.h                  \
258         hook.h                  \
259         hook_event.h            \
```

```
260         hook_impl.h             \
261         hwconf.h                \
262         ia.h                    \
263         iapriocntl.h            \
264         ibpart.h                \
265         id32.h                  \
266         idmap.h                 \
267         ieeefp.h                \
268         id_space.h              \
269         instance.h              \
270         int_const.h             \
271         int_fmtio.h             \
272         int_limits.h            \
273         int_types.h             \
274         inttypes.h              \
275         ioccom.h                \
276         ioctl.h                 \
277         ipc.h                   \
278         ipc_impl.h              \
279         ipc_rctl.h              \
280         ipmi.h                  \
281         isa_defs.h              \
282         iscsi_authclient.h      \
283         iscsi_authclientglue.h  \
284         iscsi_protocol.h        \
285         jioctl.h                \
286         kbd.h                   \
287         kbdreg.h                \
288         kbio.h                  \
289         kcpc.h                  \
290         kdi.h                   \
291         kdi_impl.h              \
292         kiconv.h                \
293         kiconv_big5_utf8.h      \
294         kiconv_cck_common.h     \
295         kiconv_cp950hkscs_utf8.h        \
296         kiconv_emea1.h          \
297         kiconv_emea2.h          \
298         kiconv_euckr_utf8.h     \
299         kiconv_euctw_utf8.h     \
300         kiconv_gb18030_utf8.h   \
301         kiconv_gb2312_utf8.h    \
302         kiconv_hkscs_utf8.h     \
303         kiconv_ja.h             \
304         kiconv_ja_jis_to_unicode.h      \
305         kiconv_ja_unicode_to_jis.h      \
306         kiconv_ko.h             \
307         kiconv_latin1.h         \
308         kiconv_sc.h             \
309         kiconv_tc.h             \
310         kiconv_uhc_utf8.h       \
311         kiconv_utf8_big5.h      \
312         kiconv_utf8_cp950hkscs.h        \
313         kiconv_utf8_euckr.h     \
314         kiconv_utf8_euctw.h     \
315         kiconv_utf8_gb18030.h   \
316         kiconv_utf8_gb2312.h    \
317         kiconv_utf8_hkscs.h     \
318         kiconv_utf8_uhc.h       \
319         kidmap.h                \
320         klpd.h                  \
321         klwp.h                  \
322         kmdb.h                  \
323         kmem.h                  \
324         kmem_impl.h             \
325         kobj.h                  \
```



```
326         kobj_impl.h             \
327         ksocket.h               \
328         kstat.h                 \
329         kstr.h                  \
330         ksyms.h                 \
331         ksynch.h                \
332         ldterm.h                \
333         lgrp.h                  \
334         lgrp_user.h             \
335         libc_kernel.h           \
336         link.h                  \
337         list.h                  \
338         list_impl.h             \
339         llc1.h                  \
340         loadavg.h               \
341         lock.h                  \
342         lockfs.h                \
343         lockstat.h              \
344         lofi.h                  \
345         log.h                   \
346         logindmux.h             \
347         logindmux_impl.h        \
348         lwp.h                   \
349         lwp_timer_impl.h        \
350         lwp_upimutex_impl.h     \
351         lpif.h                  \
352         mac.h                   \
353         mac_client.h            \
354         mac_client_impl.h       \
355         mac_ether.h             \
356         mac_flow.h              \
357         mac_flow_impl.h         \
358         mac_impl.h              \
359         mac_provider.h          \
360         mac_soft_ring.h         \
361         mac_stat.h              \
362         machelf.h               \
363         map.h                   \
364         md4.h                   \
365         md5.h                   \
366         md5_consts.h            \
367         mdi_impldefs.h          \
368         mem.h                   \
369         mem_config.h            \
370         memlist.h               \
371         mkdev.h                 \
372         mhd.h                   \
373         mii.h                   \
374         miiregs.h               \
375         mixer.h                 \
376         mman.h                  \
377         mmapobj.h               \
378         mntent.h                \
379         mntio.h                 \
380         mnttab.h                \
381         modctl.h                \
382         mode.h                  \
383         model.h                 \
384         modhash.h               \
385         modhash_impl.h          \
386         mount.h                 \
387         mouse.h                 \
388         msacct.h                \
389         msg.h                   \
390         msg_impl.h              \
391         msio.h                  \
```

```
392         msreg.h                 \
393         mtio.h                  \
394         multidata.h             \
395         multidata_impl.h        \
396         mutex.h                 \
397         nbmlock.h               \
398         ndifm.h                 \
399         ndi_impldefs.h          \
400         net80211.h              \
401         net80211_crypto.h       \
402         net80211_ht.h           \
403         net80211_proto.h        \
404         netconfig.h             \
405         neti.h                  \
406         netstack.h              \
407         nexusdefs.h             \
408         note.h                  \
409         nvpair.h                \
410         nvpair_impl.h           \
411         objfs.h                 \
412         objfs_impl.h            \
413         ontrap.h                \
414         open.h                  \
415         openpromio.h            \
416         panic.h                 \
417         param.h                 \
418         pathconf.h              \
419         pathname.h              \
420         pattr.h                 \
421         queue.h                 \
422         serializer.h            \
423         pbio.h                  \
424         pccard.h                \
425         pci.h                   \
426         pcie.h                  \
427         pci_impl.h              \
428         pci_tools.h             \
429         pcmcia.h                \
430         pctypes.h               \
431         pfmod.h                 \
432         pg.h                    \
433         pghw.h                  \
434         physmem.h               \
435         pkp_hash.h              \
436         pm.h                    \
437         policy.h                \
438         poll.h                  \
439         poll_impl.h             \
440         pool.h                  \
441         pool_impl.h             \
442         pool_pset.h             \
443         port.h                  \
444         port_impl.h             \
445         port_kernel.h           \
446         portif.h                \
447         ppmio.h                 \
448         pppt_ic_if.h            \
449         pppt_ioctl.h            \
450         priocntl.h              \
451         priv.h                  \
452         priv_impl.h             \
453         prnio.h                 \
454         proc.h                  \
455         processor.h             \
456         procfs.h                \
457         procset.h               \
```



```
458         project.h               \
459         protosw.h               \
460         prsystm.h               \
461         pset.h                  \
462         pshot.h                 \
463         ptem.h                  \
464         ptms.h                  \
465         ptyvar.h                \
466         raidioctl.h             \
467         ramdisk.h               \
468         random.h                \
469         rctl.h                  \
470         rctl_impl.h             \
471         rds.h                   \
472         reboot.h                \
473         refstr.h                \
474         refstr_impl.h           \
475         resource.h              \
476         rlioctl.h               \
477         rt.h                    \
478         rtpriocntl.h            \
479         rwlock.h                \
480         rwlock_impl.h           \
481         rwstlock.h              \
482         sad.h                   \
483         schedctl.h              \
484         sdt.h                   \
485         select.h                \
486         sem.h                   \
487         sem_impl.h              \
488         sema_impl.h             \
489         semaphore.h             \
490         sendfile.h              \
491         ser_sync.h              \
492         session.h               \
493         sha1.h                  \
494         sha1_consts.h           \
495         sha2.h                  \
496         sha2_consts.h           \
497         share.h                 \
498         shm.h                   \
499         shm_impl.h              \
500         sid.h                   \
501         siginfo.h               \
502         signal.h                \
503         sleepq.h                \
504         smbios.h                \
505         smbios_impl.h           \
506         sobject.h               \
507         socket.h                \
508         socket_impl.h           \
509         socket_proto.h          \
510         socketvar.h             \
511         sockfilter.h            \
512         sockio.h                \
513         soundcard.h             \
514         squeue.h                \
515         squeue_impl.h           \
516         srn.h                   \
517         sservice.h              \
518         stat.h                  \
519         statfs.h                \
520         statvfs.h               \
521         stdbool.h               \
522         stdint.h                \
523         stermio.h               \
```

```
524         stmf.h                  \
525         stmf_defines.h          \
526         stmf_ioctl.h            \
527         stmf_sbd_ioctl.h        \
528         stream.h                \
529         strft.h                 \
530         strlog.h                \
531         strmdep.h               \
532         stropts.h               \
533         strredir.h              \
534         strstat.h               \
535         strsubr.h               \
536         strsun.h                \
537         strtty.h                \
538         sunddi.h                \
539         sunldi.h                \
540         sunldi_impl.h           \
541         sunmdi.h                \
542         sunndi.h                \
543         sunos_dhcp_class.h      \
544         sunpm.h                 \
545         suntpi.h                \
546         suntty.h                \
547         swap.h                  \
548         synch.h                 \
549         sysdc.h                 \
550         sysdc_impl.h            \
551         syscall.h               \
552         sysconf.h               \
553         sysconfig.h             \
554         sysevent.h              \
555         sysevent_impl.h         \
556         sysinfo.h               \
557         syslog.h                \
558         sysmacros.h             \
559         sysmsg_impl.h           \
560         systeminfo.h            \
561         systm.h                 \
562         task.h                  \
563         taskq.h                 \
564         taskq_impl.h            \
565         t_kuser.h               \
566         t_lock.h                \
567         telioctl.h              \
568         termio.h                \
569         termios.h               \
570         termiox.h               \
571         thread.h                \
572         ticlts.h                \
573         ticots.h                \
574         ticotsord.h             \
575         tihdr.h                 \
576         time.h                  \
577         time_impl.h             \
578         time_std_impl.h         \
579         timeb.h                 \
580         timer.h                 \
581         times.h                 \
582         timex.h                 \
583         timod.h                 \
584         tirdwr.h                \
585         tiuser.h                \
586         tl.h                    \
587         tnf.h                   \
588         tnf_com.h               \
589         tnf_probe.h             \
```



```
590         tnf_writer.h            \
591         todio.h                 \
592         tpicommon.h             \
593         ts.h                    \
594         tspriocntl.h            \
595         ttcompat.h              \
596         ttold.h                 \
597         tty.h                   \
598         ttychars.h              \
599         ttydev.h                \
600         tuneable.h              \
601         turnstile.h             \
602         types.h                 \
603         types32.h               \
604         tzfile.h                \
605         u8_textprep.h           \
606         u8_textprep_data.h      \
607         uadmin.h                \
608         ucred.h                 \
609         uio.h                   \
610         ulimit.h                \
611         un.h                    \
612         unistd.h                \
613         user.h                  \
614         ustat.h                 \
615         utime.h                 \
616         utsname.h               \
617         utssys.h                \
618         uuid.h                  \
619         va_impl.h               \
620         va_list.h               \
621         var.h                   \
622         varargs.h               \
623         vfs.h                   \
624         vfs_opreg.h             \
625         vfstab.h                \
626         vgareg.h                \
627         videodev2.h             \
628         visual_io.h             \
629         vlan.h                  \
630         vm.h                    \
631         vm_usage.h              \
632         vmem.h                  \
633         vmem_impl.h             \
634         vmsystm.h               \
635         vnic.h                  \
636         vnic_impl.h             \
637         vnode.h                 \
638         vscan.h                 \
639         vtoc.h                  \
640         vtrace.h                \
641         vuid_event.h            \
642         vuid_wheel.h            \
643         vuid_queue.h            \
644         vuid_state.h            \
645         vuid_store.h            \
646         wait.h                  \
647         waitq.h                 \
648         wanboot_impl.h          \
649         watchpoint.h            \
650         winlockio.h             \
651         zcons.h                 \
652         zone.h                  \
653         xti_inet.h              \
654         xti_osi.h               \
655         xti_xtiopt.h            \
```

```
656         zmod.h

658 HDRS=                           \
659         $(GENHDRS)              \
660         $(CHKHDRS)

662 AUDIOHDRS=                       \
663         ac97.h                  \
664         audio_common.h          \
665         audio_driver.h          \
666         audio_oss.h             \
667         g711.h

669 AVHDRS=                          \
670         iec61883.h

672 BSCHDRS=                         \
673         bscbus.h                \
674         bscv_impl.h             \
675         lom_ebuscodes.h         \
676         lom_io.h                \
677         lom_priv.h              \
678         lombus.h

680 MDESCHDRS=                       \
681         mdesc.h                 \
682         mdesc_impl.h

684 CPUDRVHDRS=                      \
685         cpudrv.h

687 CRYPTOHDRS=                      \
688         elfsign.h               \
689         ioctl.h                 \
690         ioctladmin.h            \
691         common.h                \
692         impl.h                  \
693         spi.h                   \
694         api.h                   \
695         ops_impl.h              \
696         sched_impl.h

698 DCAMHDRS=                        \
699         dcam1394_io.h

701 IBHDRS=                 \
702         ib_types.h      \
703         ib_pkt_hdrs.h

705 IBTLHDRS=               \
706         ibtl_types.h    \
707         ibtl_status.h   \
708         ibti.h          \
709         ibti_cm.h       \
710         ibci.h          \
711         ibti_common.h   \
712         ibvti.h         \
713         ibtl_ci_types.h

715 IBTLIMPLHDRS=           \
716         ibtl_util.h

718 IBNEXHDRS=              \
719         ibnex_devctl.h

721 IBMFHDRS=               \
```



```
722         ibmf.h          \
723         ibmf_msg.h      \
724         ibmf_saa.h      \
725         ibmf_utils.h

727 IBMGTHDRS=              \
728         ib_dm_attr.h    \
729         ib_mad.h        \
730         sm_attr.h       \
731         sa_recs.h

733 IBDHDRS=                \
734         ibd.h

736 OFHDRS=                 \
737         ofa_solaris.h   \
738         ofed_kernel.h

740 RDMAHDRS=               \
741         ib_addr.h       \
742         ib_user_mad.h   \
743         ib_user_sa.h    \
744         ib_user_verbs.h \
745         ib_verbs.h      \
746         rdma_cm.h       \
747         rdma_user_cm.h

749 SOL_UVERBSHDRS=         \
750         sol_uverbs.h    \
751         sol_uverbs2ucma.h       \
752         sol_uverbs_comp.h       \
753         sol_uverbs_hca.h        \
754         sol_uverbs_qp.h         \
755         sol_uverbs_event.h

757 SOL_UMADHDRS=           \
758         sol_umad.h

760 SOL_UCMAHDRS=           \
761         sol_ucma.h      \
762         sol_rdma_user_cm.h

764 SOL_OFSHDRS=            \
765         sol_cma.h       \
766         sol_ib_cma.h    \
767         sol_ofs_common.h        \
768         sol_kverb_impl.h

770 TAVORHDRS=             \
771         tavor_ioctl.h

773 HERMONHDRS=            \
774         hermon_ioctl.h

776 MLNXHDRS=              \
777         mlnx_umap.h

779 IDMHDRS=               \
780         idm.h           \
781         idm_impl.h      \
782         idm_so.h        \
783         idm_text.h      \
784         idm_transport.h \
785         idm_conn_sm.h

787 ISCSITHDRS=             \
```

```
788         radius_packet.h         \
789         radius_protocol.h       \
790         chap.h                  \
791         isns_protocol.h         \
792         iscsi_if.h              \
793         iscsit_common.h

795 ISOHDRS=                       \
796         signal_iso.h

798 DERIVED_LVMHDRS=       \
799         md_mdiox.h              \
800         md_basic.h              \
801         mdmed.h                 \
802         md_mhdx.h               \
803         mdmn_commd.h

805 LVMHDRS=                       \
806         md_convert.h            \
807         md_crc.h                \
808         md_hotspares.h          \
809         md_mddb.h               \
810         md_mirror.h             \
811         md_mirror_shared.h      \
812         md_names.h              \
813         md_notify.h             \
814         md_raid.h               \
815         md_rename.h             \
816         md_sp.h                 \
817         md_stripe.h             \
818         md_trans.h              \
819         mdio.h                  \
820         mdvar.h

822 ALL_LVMHDRS=                   \
823         $(LVMHDRS)              \
824         $(DERIVED_LVMHDRS)

826 FMHDRS=                        \
827         protocol.h              \
828         util.h

830 FMFSHDRS=                      \
831         zfs.h

833 FMIOHDRS=                      \
834         ddi.h                   \
835         disk.h                  \
836         pci.h                   \
837         scsi.h                  \
838         sun4upci.h              \
839         opl_mc_fm.h

841 FSHDRS=                        \
842         autofs.h                \
843         cachefs_dir.h           \
844         cachefs_dlog.h          \
845         cachefs_filegrp.h       \
846         cachefs_fs.h            \
847         cachefs_fscache.h       \
848         cachefs_ioctl.h         \
849         cachefs_log.h           \
850         decomp.h                \
851         dv_node.h               \
852         sdev_impl.h             \
853         fifonode.h              \
```



```
854         hsfs_isospec.h          \
855         hsfs_node.h             \
856         hsfs_rrip.h             \
857         hsfs_spec.h             \
858         hsfs_susp.h             \
859         lofs_info.h             \
860         lofs_node.h             \
861         mntdata.h               \
862         namenode.h              \
863         pc_dir.h                \
864         pc_fs.h                 \
865         pc_label.h              \
866         pc_node.h               \
867         pxfs_ki.h               \
868         snode.h                 \
869         swapnode.h              \
870         tmp.h                   \
871         tmpnode.h               \
872         udf_inode.h             \
873         udf_volume.h            \
874         ufs_acl.h               \
875         ufs_bio.h               \
876         ufs_filio.h             \
877         ufs_fs.h                \
878         ufs_fsdir.h             \
879         ufs_inode.h             \
880         ufs_lockfs.h            \
881         ufs_log.h               \
882         ufs_mount.h             \
883         ufs_panic.h             \
884         ufs_prot.h              \
885         ufs_quota.h             \
886         ufs_snap.h              \
887         ufs_trans.h             \
888         zfs.h                   \
889         zut.h

891 PCMCIAHDRS=             \
892         pcata.h         \
893         pcser_conf.h    \
894         pcser_io.h      \
895         pcser_reg.h     \
896         pcser_manuspec.h \
897         pcser_var.h

899 SCSIHDRS=               \
900         scsi.h          \
901         scsi_address.h  \
902         scsi_ctl.h      \
903         scsi_fm.h       \
904         scsi_params.h   \
905         scsi_pkt.h      \
906         scsi_resource.h \
907         scsi_types.h    \
908         scsi_watch.h

910 SCSICONFHDRS=           \
911         autoconf.h      \
912         device.h

914 SCSIGENHDRS=            \
915         commands.h      \
916         dad_mode.h      \
917         inquiry.h       \
918         message.h       \
919         mode.h          \
```

```
920         persist.h       \
921         sense.h         \
922         sff_frames.h    \
923         smp_frames.h    \
924         status.h

926 SCSIIMPLHDRS=           \
927         commands.h      \
928         inquiry.h       \
929         mode.h          \
930         scsi_reset_notify.h \
931         scsi_sas.h      \
932         sense.h         \
933         services.h      \
934         smp_transport.h \
935         spc3_types.h    \
936         status.h        \
937         transport.h     \
938         types.h         \
939         uscsi.h         \
940         usmp.h

942 SCSITARGETSHDRS=        \
943         ses.h           \
944         sesio.h         \
945         sgendef.h       \
946         stdef.h         \
947         sddef.h         \
948         smp.h

950 SCSIADHDRS=

952 SCSICADHDRS=

954 SCSIISCSIHDRS=          \
955         iscsi_door.h    \
956         iscsi_if.h

958 SCSIVHCIHDRS=           \
959         scsi_vhci.h     \
960         mpapi_impl.h    \
961         mpapi_scsi_vhci.h

963 SDCARDHDRS=             \
964         sda.h           \
965         sda_impl.h      \
966         sda_ioctl.h

968 FC4HDRS=                       \
969         fc_transport.h  \
970         linkapp.h       \
971         fc.h            \
972         fcp.h           \
973         fcal_transport.h \
974         fcal.h          \
975         fcal_linkapp.h  \
976         fcio.h

978 FCHDRS=                \
979         fc.h            \
980         fcio.h          \
981         fc_types.h      \
982         fc_appif.h

984 FCIMPLHDRS=            \
985         fc_error.h      \
```



```
986         fcph.h

988 FCULPHDRS=             \
989         fcp_util.h      \
990         fcsm.h

992 SATAGENHDRS=           \
993         sata_hba.h      \
994         sata_defs.h     \
995         sata_cfgadm.h

997 SYSEVENTHDRS=          \
998         ap_driver.h     \
999         dev.h           \
1000        domain.h        \
1001        dr.h            \
1002        env.h           \
1003        eventdefs.h     \
1004        ipmp.h          \
1005        pwrctl.h        \
1006        svm.h           \
1007        vrrp.h

1009 CONTRACTHDRS=          \
1010        process.h       \
1011        process_impl.h  \
1012        device.h        \
1013        device_impl.h

1015 USBHDRS=        \
1016        usba.h  \
1017        usbai.h

1019 UWBHDRS=        \
1020        uwb.h   \
1021        uwbai.h

1023 UWBAHDRS=       \
1024        uwba.h

1026 USBAUDHDRS=           \
1027        usb_audio.h

1029 USBHUBDHDRS=          \
1030        hub.h           \
1031        hubd_impl.h

1033 USBHIDHDRS=           \
1034        hid.h

1036 USBHWARCHDRS=         \
1037        hwarc.h

1039 USBMSHDRS=            \
1040        usb_bulkonly.h  \
1041        usb_cbi.h

1043 USBPRNHDRS=          \
1044        usb_printer.h

1046 USBCDCHDRS=          \
1047        usb_cdc.h

1049 USBVIDHDRS=          \
1050        usbvc.h
```

```
1052 USBWCMHDRS=             \
1053         usbwcm.h

1055 UGENHDRS=           \
1056         usb_ugen.h

1058 HOTPLUGHDRS=    \
1059         hpcsvc.h        \
1060         hpctrl.h

1062 HOTPLUGPCIHDRS= \
1063         pcicfg.h        \
1064         pcihp.h

1066 RSMHDRS= \
1067         rsm.h   \
1068         rsm_common.h    \
1069         rsmapi_common.h \
1070         rsmpi.h \
1071         rsmpi_driver.h  \
1072         rsmka_path_int.h

1074 TSOLHDRS=               \
1075         label.h         \
1076         label_macro.h   \
1077         priv.h          \
1078         tndb.h          \
1079         tsyscall.h

1081 I1394HDRS=              \
1082         cmd1394.h       \
1083         id1394.h        \
1084         ieee1212.h      \
1085         ieee1394.h      \
1086         ixl1394.h       \
1087         s1394_impl.h    \
1088         t1394.h

1090 # "cmdk" headers used on sparc
1091 SDKTPHDRS=              \
1092         dadkio.h        \
1093         fdisk.h

1095 # "cmdk" headers used on i386
1096 DKTPHDRS=               \
1097         altsctr.h       \
1098         bbh.h           \
1099         cm.h            \
1100         cmdev.h         \
1101         cmdk.h          \
1102         cmpkt.h         \
1103         controller.h    \
1104         dadev.h         \
1105         dadk.h          \
1106         dadkio.h        \
1107         fctypes.h       \
1108         fdisk.h         \
1109         flowctrl.h      \
1110         gda.h           \
1111         quetypes.h      \
1112         queue.h         \
1113         tgcom.h         \
1114         tgdk.h

1116 # "pc" header files used on i386
1117 PCHDRS=                 \
```



```
1118         avintr.h        \
1119         dma_engine.h    \
1120         i8272A.h        \
1121         pcic_reg.h      \
1122         pcic_var.h      \
1123         pic.h           \
1124         pit.h           \
1125         rtc.h

1127 NXGEHDRS=                       \
1128         nxge.h                  \
1129         nxge_common.h           \
1130         nxge_common_impl.h      \
1131         nxge_defs.h             \
1132         nxge_hw.h               \
1133         nxge_impl.h             \
1134         nxge_ipp.h              \
1135         nxge_ipp_hw.h           \
1136         nxge_mac.h              \
1137         nxge_mac_hw.h           \
1138         nxge_fflp.h             \
1139         nxge_fflp_hw.h          \
1140         nxge_mii.h              \
1141         nxge_rxdma.h            \
1142         nxge_rxdma_hw.h         \
1143         nxge_txc.h              \
1144         nxge_txc_hw.h           \
1145         nxge_txdma.h            \
1146         nxge_txdma_hw.h         \
1147         nxge_virtual.h          \
1148         nxge_espc.h

1150 include Makefile.syshdrs

1152 dcam/%.check:   dcam/%.h
1153         $(DOT_H_CHECK)

1155 CHECKHDRS=                                                      \
1156         $($(MACH)_HDRS:%.h=%.check)                             \
1157         $(AUDIOHDRS:%.h=audio/%.check)                          \
1158         $(AVHDRS:%.h=av/%.check)                                \
1159         $(BSCHDRS:%.h=%.check)                                  \
1160         $(CHKHDRS:%.h=%.check)                                  \
1161         $(CPUDRVHDRS:%.h=%.check)                               \
1162         $(CRYPTOHDRS:%.h=crypto/%.check)                        \
1163         $(DCAMHDRS:%.h=dcam/%.check)                            \
1164         $(FC4HDRS:%.h=fc4/%.check)                              \
1165         $(FCHDRS:%.h=fibre-channel/%.check)                     \
1166         $(FCIMPLHDRS:%.h=fibre-channel/impl/%.check)            \
1167         $(FCULPHDRS:%.h=fibre-channel/ulp/%.check)              \
1168         $(IBHDRS:%.h=ib/%.check)                                \
1169         $(IBDHDRS:%.h=ib/clients/ibd/%.check)                   \
1170         $(IBTLHDRS:%.h=ib/ibtl/%.check)                         \
1171         $(IBTLIMPLHDRS:%.h=ib/ibtl/impl/%.check)                \
1172         $(IBNEXHDRS:%.h=ib/ibnex/%.check)       \
1173         $(IBMGTHDRS:%.h=ib/mgt/%.check)                         \
1174         $(IBMFHDRS:%.h=ib/mgt/ibmf/%.check)                     \
1175         $(OFHDRS:%.h=ib/clients/of/%.check)                                     \
1176         $(RDMAHDRS:%.h=ib/clients/of/rdma/%.check)                              \
1177         $(SOL_UVERBSHDRS:%.h=ib/clients/of/sol_uverbs/%.check)  \
1178         $(SOL_UCMAHDRS:%.h=ib/clients/of/sol_ucma/%.check)      \
1179         $(SOL_OFSHDRS:%.h=ib/clients/of/sol_ofs/%.check)        \
1180         $(TAVORHDRS:%.h=ib/adapters/tavor/%.check)      \
1181         $(HERMONHDRS:%.h=ib/adapters/hermon/%.check)    \
1182         $(MLNXHDRS:%.h=ib/adapters/%.check)             \
1183         $(IDMHDRS:%.h=idm/%.check)                      \
```

```
1184         $(ISCSIHDRS:%.h=iscsi/%.check)                 \
1185         $(ISCSITHDRS:%.h=iscsit/%.check)                       \
1186         $(ISOHDRS:%.h=iso/%.check)                     \
1187         $(FMHDRS:%.h=fm/%.check)                       \
1188         $(FMFSHDRS:%.h=fm/fs/%.check)                  \
1189         $(FMIOHDRS:%.h=fm/io/%.check)                  \
1190         $(FSHDRS:%.h=fs/%.check)                       \
1191         $(LVMHDRS:%.h=lvm/%.check)                     \
1192         $(PCMCIAHDRS:%.h=pcmcia/%.check)               \
1193         $(SCSIHDRS:%.h=scsi/%.check)                   \
1194         $(SCSIADHDRS:%.h=scsi/adapters/%.check)        \
1195         $(SCSICONFHDRS:%.h=scsi/conf/%.check)          \
1196         $(SCSIIMPLHDRS:%.h=scsi/impl/%.check)          \
1197         $(SCSIISCSIHDRS:%.h=scsi/adapters/%.check)     \
1198         $(SCSIGENHDRS:%.h=scsi/generic/%.check)        \
1199         $(SCSITARGETSHDRS:%.h=scsi/targets/%.check)    \
1200         $(SCSIVHCIHDRS:%.h=scsi/adapters/%.check)      \
1201         $(SATAGENHDRS:%.h=sata/%.check)                \
1202         $(SDCARDHDRS:%.h=sdcard/%.check)               \
1203         $(SYSEVENTHDRS:%.h=sysevent/%.check)           \
1204         $(CONTRACTHDRS:%.h=contract/%.check)           \
1205         $(USBAUDHDRS:%.h=usb/clients/audio/%.check)   \
1206         $(USBHUBDHDRS:%.h=usb/hubd/%.check)            \
1207         $(USBHIDHDRS:%.h=usb/clients/hid/%.check)      \
1208         $(USBHWARCHDRS:%.h=usb/clients/hwarc/%.check)   \
1209         $(USBMSHDRS:%.h=usb/clients/mass_storage/%.check)       \
1210         $(USBPRNHDRS:%.h=usb/clients/printer/%.check)   \
1211         $(USBCDCHDRS:%.h=usb/clients/usbcdc/%.check)    \
1212         $(USBVIDHDRS:%.h=usb/clients/video/usbvc/%.check)   \
1213         $(USBWCMHDRS:%.h=usb/clients/usbinput/usbwcm/%.check)   \
1214         $(UGENHDRS:%.h=usb/clients/ugen/%.check)   \
1215         $(USBHDRS:%.h=usb/%.check)                     \
1216         $(UWBHDRS:%.h=uwb/%.check)                     \
1217         $(UWBAHDRS:%.h=uwb/uwba/%.check)                    \
1218         $(I1394HDRS:%.h=1394/%.check)                  \
1219         $(RSMHDRS:%.h=rsm/%.check)                     \
1220         $(TSOLHDRS:%.h=tsol/%.check)                   \
1221         $(NXGEHDRS:%.h=nxge/%.check)


1224 .KEEP_STATE:

1226 .PARALLEL:                      \
1227         $(CHECKHDRS)            \
1228         $(ROOTHDRS)             \
1229         $(ROOTAUDHDRS)          \
1230         $(ROOTAVHDRS)           \
1231         $(ROOTCRYPTOHDRS)       \
1232         $(ROOTDCAMHDRS)         \
1233         $(ROOTISOHDRS)          \
1234         $(ROOTIDMHDRS)          \
1235         $(ROOTISCSIHDRS)        \
1236         $(ROOTISCSITHDRS)       \
1237         $(ROOTFC4HDRS)          \
1238         $(ROOTFCHDRS)           \
1239         $(ROOTFCIMPLHDRS)       \
1240         $(ROOTFCULPHDRS)        \
1241         $(ROOTFMHDRS)           \
1242         $(ROOTFMIOHDRS)         \
1243         $(ROOTFMFSHDRS)         \
1244         $(ROOTFSHDRS)           \
1245         $(ROOTIBDHDRS)          \
1246         $(ROOTIBHDRS)           \
1247         $(ROOTIBTLHDRS)         \
1248         $(ROOTIBTLIMPLHDRS)     \
1249         $(ROOTIBNEXHDRS)        \
```



```
1250         $(ROOTIBMGTHDRS)        \
1251         $(ROOTIBMFHDRS)         \
1252         $(ROOTOFHDRS)           \
1253         $(ROOTRDMAHDRS)         \
1254         $(ROOTSOL_OFSHDRS)      \
1255         $(ROOTSOL_UMADHDRS)     \
1256         $(ROOTSOL_UVERBSHDRS)   \
1257         $(ROOTSOL_UCMAHDRS)     \
1258         $(ROOTTAVORHDRS)        \
1259         $(ROOTHERMONHDRS)       \
1260         $(ROOTMLNXHDRS)         \
1261         $(ROOTLVMHDRS)          \
1262         $(ROOTPCMCIAHDRS)       \
1263         $(ROOTSCSIHDRS)         \
1264         $(ROOTSCSIADHDRS)       \
1265         $(ROOTSCSICONFHDRS)     \
1266         $(ROOTSCSIISCSIHDRS)    \
1267         $(ROOTSCSIGENHDRS)      \
1268         $(ROOTSCSIIMPLHDRS)     \
1269         $(ROOTSCSIVHCIHDRS)     \
1270         $(ROOTSDCARDHDRS)       \
1271         $(ROOTSYSEVENTHDRS)     \
1272         $(ROOTCONTRACTHDRS)     \
1273         $(ROOTUSBHDRS)          \
1274         $(ROOTUWBHDRS)          \
1275         $(ROOTUWBAHDRS)         \
1276         $(ROOTUSBAUDHDRS)       \
1277         $(ROOTUSBHUBDHDRS)      \
1278         $(ROOTUSBHIDHDRS)       \
1279         $(ROOTUSBHRCHDRS)       \
1280         $(ROOTUSBMSHDRS)        \
1281         $(ROOTUSBPRNHDRS)       \
1282         $(ROOTUSBCDCHDRS)       \
1283         $(ROOTUSBVIDHDRS)       \
1284         $(ROOTUSBWCMHDRS)       \
1285         $(ROOTUGENHDRS)         \
1286         $(ROOT1394HDRS)         \
1287         $(ROOTHOTPLUGHDRS)      \
1288         $(ROOTHOTPLUGPCIHDRS)   \
1289         $(ROOTRSMHDRS)          \
1290         $(ROOTTSOLHDRS)         \
1291         $($(MACH)_ROOTHDRS)


1294 install_h:                      \
1295         $(ROOTDIRS)             \
1296         LVMDERIVED_H            \
1297         .WAIT                   \
1298         $(ROOTHDRS)             \
1299         $(ROOTAUDHDRS)          \
1300         $(ROOTAVHDRS)           \
1301         $(ROOTCRYPTOHDRS)       \
1302         $(ROOTDCAMHDRS)         \
1303         $(ROOTISOHDRS)          \
1304         $(ROOTIDMHDRS)          \
1305         $(ROOTISCSIHDRS)        \
1306         $(ROOTISCSITHDRS)       \
1307         $(ROOTFC4HDRS)          \
1308         $(ROOTFCHDRS)           \
1309         $(ROOTFCIMPLHDRS)       \
1310         $(ROOTFCULPHDRS)        \
1311         $(ROOTFMHDRS)           \
1312         $(ROOTFMFSHDRS)         \
1313         $(ROOTFMIOHDRS)         \
1314         $(ROOTFSHDRS)           \
1315         $(ROOTIBDHDRS)          \
```

```
1316         $(ROOTIBHDRS)           \
1317         $(ROOTIBTLHDRS)         \
1318         $(ROOTIBTLIMPLHDRS)     \
1319         $(ROOTIBNEXHDRS)        \
1320         $(ROOTIBMGTHDRS)        \
1321         $(ROOTIBMFHDRS)         \
1322         $(ROOTOFHDRS)           \
1323         $(ROOTRDMAHDRS)         \
1324         $(ROOTSOL_OFSHDRS)      \
1325         $(ROOTSOL_UMADHDRS)     \
1326         $(ROOTSOL_UVERBSHDRS)   \
1327         $(ROOTSOL_UCMAHDRS)     \
1328         $(ROOTTAVORHDRS)        \
1329         $(ROOTHERMONHDRS)       \
1330         $(ROOTMLNXHDRS)         \
1331         $(ROOTLVMHDRS)          \
1332         $(ROOTPCMCIAHDRS)       \
1333         $(ROOTSCSIHDRS)         \
1334         $(ROOTSCSIADHDRS)       \
1335         $(ROOTSCSIISCSIHDRS)    \
1336         $(ROOTSCSICONFHDRS)     \
1337         $(ROOTSCSIGENHDRS)      \
1338         $(ROOTSCSIIMPLHDRS)     \
1339         $(ROOTSCSIVHCIHDRS)     \
1340         $(ROOTSDCARDHDRS)       \
1341         $(ROOTSYSEVENTHDRS)     \
1342         $(ROOTCONTRACTHDRS)     \
1343         $(ROOTUWBHDRS)          \
1344         $(ROOTUWBAHDRS)         \
1345         $(ROOTUSBHDRS)          \
1346         $(ROOTUSBAUDHDRS)       \
1347         $(ROOTUSBHUBDHDRS)      \
1348         $(ROOTUSBHIDHDRS)       \
1349         $(ROOTUSBHRCHDRS)       \
1350         $(ROOTUSBMSHDRS)        \
1351         $(ROOTUSBPRNHDRS)       \
1352         $(ROOTUSBCDCHDRS)       \
1353         $(ROOTUSBVIDHDRS)       \
1354         $(ROOTUSBWCMHDRS)       \
1355         $(ROOTUGENHDRS)         \
1356         $(ROOT1394HDRS)         \
1357         $(ROOTHOTPLUGHDRS)      \
1358         $(ROOTHOTPLUGPCIHDRS)   \
1359         $(ROOTRSMHDRS)          \
1360         $(ROOTTSOLHDRS)         \
1361         $($(MACH)_ROOTHDRS)

1363 all_h: $(GENHDRS)

1365 priv_const.h: $(PRIVS_AWK) $(PRIVS_DEF)
1366         $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v privhfile=$@

1368 priv_names.h: $(PRIVS_AWK) $(PRIVS_DEF)
1369         $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v pubhfile=$@

1371 usb/usbdevs.h: $(USBDEVS_AWK) $(USBDEVS_DATA)
1372         $(NAWK) -f $(USBDEVS_AWK) $(USBDEVS_DATA) -H > $@

1374 LVMDERIVED_H:
1375         cd $(SRC)/uts/common/sys/lvm; pwd; $(MAKE)

1377 clean:
1378         $(RM) $(GENHDRS)

1380 clobber:        clean
```



```
1382 check:  $(CHECKHDRS)

1384 FRC:

1386 # EXPORT DELETE START
1387 EXPORT_SRC:
1388         $(RM) wanboot_impl.h+ Makefile+
1389         sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
1390                 < wanboot_impl.h > wanboot_impl.h+
1391         $(MV) wanboot_impl.h+ wanboot_impl.h
1392         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
1393                 < Makefile > Makefile+
1394         $(RM) Makefile
1395         $(MV) Makefile+ Makefile
1396         $(CHMOD) 444 Makefile wanboot_impl.h
1397 # EXPORT DELETE END
```

```
**********************************************************
    2168 Thu Jul 11 01:30:11 2013
new/usr/src/uts/common/sys/wanboot_impl.h
first pass
**********************************************************
```

```
/*
 * 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 2002-2003 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#ifndef _SYS_WANBOOT_IMPL_H
#define _SYS_WANBOOT_IMPL_H

#pragma ident   "%Z%%M% %I%     %E% SMI"

#include <sys/types.h>
/* EXPORT DELETE START */
#include <aes.h>
#include <des3.h>
#include <hmac_sha1.h>
/* EXPORT DELETE END */

#ifdef  __cplusplus
extern "C" {
#endif

/*
 * PKCS12 passphrase used by WAN boot
 */
#define WANBOOT_PASSPHRASE      "boy with goldfish"

/*
 * Key names used by OBP.
 */
#define WANBOOT_DES3_KEY_NAME           "wanboot-3des"
#define WANBOOT_AES_128_KEY_NAME        "wanboot-aes"
#define WANBOOT_HMAC_SHA1_KEY_NAME      "wanboot-hmac-sha1"
#define WANBOOT_MAXKEYNAMELEN           sizeof (WANBOOT_HMAC_SHA1_KEY_NAME)

#define WANBOOT_MAXKEYLEN       1024    /* sized for RSA */

/* EXPORT DELETE START */
#define WANBOOT_MAXBLOCKLEN     AES_BLOCK_SIZE
#define WANBOOT_HMAC_KEY_SIZE   20      /* size of key we use for HMAC SHA-1 */
/* EXPORT DELETE END */
```



```
struct wankeyio {
        char    wk_keyname[WANBOOT_MAXKEYNAMELEN];
        uint_t  wk_keysize;
        union {
/* EXPORT DELETE START */
                char    hmac_sha1_key[WANBOOT_HMAC_KEY_SIZE];
                char    des3key[DES3_KEY_SIZE];
                char    aeskey[AES_128_KEY_SIZE];
/* EXPORT DELETE END */
                char    key[WANBOOT_MAXKEYLEN];
        } wk_u;
};

/* EXPORT DELETE START */
#define wk_hmac_sha1_key        wk_u.hmac_sha1_key
#define wk_3des_key             wk_u.3des_key
#define wk_aes_key              wk_u.aeskey
/* EXPORT DELETE END */

#define WANBOOT_SETKEY          (('W' << 24) | ('A' << 16) | ('N' << 8) | 0)

#ifdef  __cplusplus
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   20693 Thu Jul 11 01:30:11 2013
new/usr/src/uts/common/syscall/fcntl.c
onc plus-be-gone
**********************************************************
```

```
/*
 * 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
 */
/* ONC_PLUS EXTRACT START */
/*
 * Copyright (c) 1994, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
 */

/*      Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989 AT&T     */
/*        All Rights Reserved   */

/*
 * Portions of this source code were derived from Berkeley 4.3 BSD
 * under license from the Regents of the University of California.
 */


/* ONC_PLUS EXTRACT END */

#include <sys/param.h>
#include <sys/isa_defs.h>
#include <sys/types.h>
#include <sys/sysmacros.h>
#include <sys/systm.h>
#include <sys/errno.h>
#include <sys/fcntl.h>
/* ONC_PLUS EXTRACT START */
#include <sys/flock.h>
/* ONC_PLUS EXTRACT END */
#include <sys/vnode.h>
#include <sys/file.h>
#include <sys/mode.h>
#include <sys/proc.h>
#include <sys/filio.h>
#include <sys/share.h>
#include <sys/debug.h>
#include <sys/rctl.h>
#include <sys/nbmlock.h>

#include <sys/cmn_err.h>

/* ONC_PLUS EXTRACT START */
```



```
static int flock_check(vnode_t *, flock64_t *, offset_t, offset_t);
static int flock_get_start(vnode_t *, flock64_t *, offset_t, u_offset_t *);
static void fd_too_big(proc_t *);

/*
 * File control.
 */
int
fcntl(int fdes, int cmd, intptr_t arg)
{
        int iarg;
        int error = 0;
        int retval;
        proc_t *p;
        file_t *fp;
        vnode_t *vp;
        u_offset_t offset;
        u_offset_t start;
        struct vattr vattr;
        int in_crit;
        int flag;
        struct flock sbf;
        struct flock64 bf;
        struct o_flock obf;
        struct flock64_32 bf64_32;
        struct fshare fsh;
        struct shrlock shr;
        struct shr_locowner shr_own;
        offset_t maxoffset;
        model_t datamodel;
        int fdres;

#if defined(_ILP32) && !defined(lint) && defined(_SYSCALL32)
        ASSERT(sizeof (struct flock) == sizeof (struct flock32));
        ASSERT(sizeof (struct flock64) == sizeof (struct flock64_32));
#endif
#if defined(_LP64) && !defined(lint) && defined(_SYSCALL32)
        ASSERT(sizeof (struct flock) == sizeof (struct flock64_64));
        ASSERT(sizeof (struct flock64) == sizeof (struct flock64_64));
#endif

        /*
         * First, for speed, deal with the subset of cases
         * that do not require getf() / releasef().
         */
        switch (cmd) {
        case F_GETFD:
                if ((error = f_getfd_error(fdes, &flag)) == 0)
                        retval = flag;
                goto out;

        case F_SETFD:
                error = f_setfd_error(fdes, (int)arg);
                retval = 0;
                goto out;

        case F_GETFL:
                if ((error = f_getfl(fdes, &flag)) == 0) {
                        retval = (flag & (FMASK | FASYNC));
                        if ((flag & (FSEARCH | FEXEC)) == 0)
                                retval += FOPEN;
                        else
                                retval |= (flag & (FSEARCH | FEXEC));
                }
                goto out;
```

```
122         case F_GETXFL:
123                 if ((error = f_getfl(fdes, &flag)) == 0) {
124                         retval = flag;
125                         if ((flag & (FSEARCH | FEXEC)) == 0)
126                                 retval += FOPEN;
127                 }
128                 goto out;

130         case F_BADFD:
131                 if ((error = f_badfd(fdes, &fdres, (int)arg)) == 0)
132                         retval = fdres;
133                 goto out;
134         }

136         /*
137          * Second, for speed, deal with the subset of cases that
138          * require getf() / releasef() but do not require copyin.
139          */
140         if ((fp = getf(fdes)) == NULL) {
141                 error = EBADF;
142                 goto out;
143         }
144         iarg = (int)arg;

146         switch (cmd) {
153 /* ONC_PLUS EXTRACT END */

147         case F_DUPFD:
148         case F_DUPFD_CLOEXEC:
149                 p = curproc;
150                 if ((uint_t)iarg >= p->p_fno_ctl) {
151                         if (iarg >= 0)
152                                 fd_too_big(p);
153                         error = EINVAL;
154                         goto done;
155                 }
156                 /*
157                  * We need to increment the f_count reference counter
158                  * before allocating a new file descriptor.
159                  * Doing it other way round opens a window for race condition
160                  * with closeandsetf() on the target file descriptor which can
161                  * close the file still referenced by the original
162                  * file descriptor.
163                  */
164                 mutex_enter(&fp->f_tlock);
165                 fp->f_count++;
166                 mutex_exit(&fp->f_tlock);
167                 if ((retval = ufalloc_file(iarg, fp)) == -1) {
168                         /*
169                          * New file descriptor can't be allocated.
170                          * Revert the reference count.
171                          */
172                         mutex_enter(&fp->f_tlock);
173                         fp->f_count--;
174                         mutex_exit(&fp->f_tlock);
175                         error = EMFILE;
176                 } else {
177                         if (cmd == F_DUPFD_CLOEXEC) {
178                                 f_setfd(retval, FD_CLOEXEC);
179                         }
180                 }
181                 goto done;

183         case F_DUP2FD_CLOEXEC:
184                 if (fdes == iarg) {
185                         error = EINVAL;
```



```
186                         goto done;
187                 }

189                 /*FALLTHROUGH*/

191         case F_DUP2FD:
192                 p = curproc;
193                 if (fdes == iarg) {
194                         retval = iarg;
195                 } else if ((uint_t)iarg >= p->p_fno_ctl) {
196                         if (iarg >= 0)
197                                 fd_too_big(p);
198                         error = EBADF;
199                 } else {
200                         /*
201                          * We can't hold our getf(fdes) across the call to
202                          * closeandsetf() because it creates a window for
203                          * deadlock: if one thread is doing dup2(a, b) while
204                          * another is doing dup2(b, a), each one will block
205                          * waiting for the other to call releasef().  The
206                          * solution is to increment the file reference count
207                          * (which we have to do anyway), then releasef(fdes),
208                          * then closeandsetf().  Incrementing f_count ensures
209                          * that fp won't disappear after we call releasef().
210                          * When closeandsetf() fails, we try avoid calling
211                          * closef() because of all the side effects.
212                          */
213                         mutex_enter(&fp->f_tlock);
214                         fp->f_count++;
215                         mutex_exit(&fp->f_tlock);
216                         releasef(fdes);
217                         if ((error = closeandsetf(iarg, fp)) == 0) {
218                                 if (cmd == F_DUP2FD_CLOEXEC) {
219                                         f_setfd(iarg, FD_CLOEXEC);
220                                 }
221                                 retval = iarg;
222                         } else {
223                                 mutex_enter(&fp->f_tlock);
224                                 if (fp->f_count > 1) {
225                                         fp->f_count--;
226                                         mutex_exit(&fp->f_tlock);
227                                 } else {
228                                         mutex_exit(&fp->f_tlock);
229                                         (void) closef(fp);
230                                 }
231                         }
232                         goto out;
233                 }
234                 goto done;

236         case F_SETFL:
237                 vp = fp->f_vnode;
238                 flag = fp->f_flag;
239                 if ((iarg & (FNONBLOCK|FNDELAY)) == (FNONBLOCK|FNDELAY))
240                         iarg &= ~FNDELAY;
241                 if ((error = VOP_SETFL(vp, flag, iarg, fp->f_cred, NULL)) ==
242                     0) {
243                         iarg &= FMASK;
244                         mutex_enter(&fp->f_tlock);
245                         fp->f_flag &= ~FMASK | (FREAD|FWRITE);
246                         fp->f_flag |= (iarg - FOPEN) & ~(FREAD|FWRITE);
247                         mutex_exit(&fp->f_tlock);
248                 }
249                 retval = 0;
250                 goto done;
251         }
```

```
253         /*
254          * Finally, deal with the expensive cases.
255          */
256         retval = 0;
257         in_crit = 0;
258         maxoffset = MAXOFF_T;
259         datamodel = DATAMODEL_NATIVE;
260 #if defined(_SYSCALL32_IMPL)
261         if ((datamodel = get_udatamodel()) == DATAMODEL_ILP32)
262                 maxoffset = MAXOFF32_T;
263 #endif

265         vp = fp->f_vnode;
266         flag = fp->f_flag;
267         offset = fp->f_offset;

269         switch (cmd) {
278 /* ONC_PLUS EXTRACT START */
270         /*
271          * The file system and vnode layers understand and implement
272          * locking with flock64 structures. So here once we pass through
273          * the test for compatibility as defined by LFS API, (for F_SETLK,
274          * F_SETLKW, F_GETLK, F_GETLKW, F_FREESP) we transform
275          * the flock structure to a flock64 structure and send it to the
276          * lower layers. Similarly in case of GETLK the returned flock64
277          * structure is transformed to a flock structure if everything fits
278          * in nicely, otherwise we return EOVERFLOW.
279          */

281         case F_GETLK:
282         case F_O_GETLK:
283         case F_SETLK:
284         case F_SETLKW:
285         case F_SETLK_NBMAND:

287                 /*
288                  * Copy in input fields only.
289                  */

291                 if (cmd == F_O_GETLK) {
292                         if (datamodel != DATAMODEL_ILP32) {
293                                 error = EINVAL;
294                                 break;
295                         }

297                         if (copyin((void *)arg, &obf, sizeof (obf))) {
298                                 error = EFAULT;
299                                 break;
300                         }
301                         bf.l_type = obf.l_type;
302                         bf.l_whence = obf.l_whence;
303                         bf.l_start = (off64_t)obf.l_start;
304                         bf.l_len = (off64_t)obf.l_len;
305                         bf.l_sysid = (int)obf.l_sysid;
306                         bf.l_pid = obf.l_pid;
307                 } else if (datamodel == DATAMODEL_NATIVE) {
308                         if (copyin((void *)arg, &sbf, sizeof (sbf))) {
309                                 error = EFAULT;
310                                 break;
311                         }
312                         /*
313                          * XXX  In an LP64 kernel with an LP64 application
314                          *      there's no need to do a structure copy here
315                          *      struct flock == struct flock64. However,
316                          *      we did it this way to avoid more conditional
```



```
317                          *      compilation.
318                          */
319                         bf.l_type = sbf.l_type;
320                         bf.l_whence = sbf.l_whence;
321                         bf.l_start = (off64_t)sbf.l_start;
322                         bf.l_len = (off64_t)sbf.l_len;
323                         bf.l_sysid = sbf.l_sysid;
324                         bf.l_pid = sbf.l_pid;
325                 }
326 #if defined(_SYSCALL32_IMPL)
327                 else {
328                         struct flock32 sbf32;
329                         if (copyin((void *)arg, &sbf32, sizeof (sbf32))) {
330                                 error = EFAULT;
331                                 break;
332                         }
333                         bf.l_type = sbf32.l_type;
334                         bf.l_whence = sbf32.l_whence;
335                         bf.l_start = (off64_t)sbf32.l_start;
336                         bf.l_len = (off64_t)sbf32.l_len;
337                         bf.l_sysid = sbf32.l_sysid;
338                         bf.l_pid = sbf32.l_pid;
339                 }
340 #endif /* _SYSCALL32_IMPL */

342                 /*
343                  * 64-bit support: check for overflow for 32-bit lock ops
344                  */
345                 if ((error = flock_check(vp, &bf, offset, maxoffset)) != 0)
346                         break;

348                 /*
349                  * Not all of the filesystems understand F_O_GETLK, and
350                  * there's no need for them to know.  Map it to F_GETLK.
351                  */
352                 if ((error = VOP_FRLOCK(vp, (cmd == F_O_GETLK) ? F_GETLK : cmd,
353                     &bf, flag, offset, NULL, fp->f_cred, NULL)) != 0)
354                         break;

356                 /*
357                  * If command is GETLK and no lock is found, only
358                  * the type field is changed.
359                  */
360                 if ((cmd == F_O_GETLK || cmd == F_GETLK) &&
361                     bf.l_type == F_UNLCK) {
362                         /* l_type always first entry, always a short */
363                         if (copyout(&bf.l_type, &((struct flock *)arg)->l_type,
364                             sizeof (bf.l_type)))
365                                 error = EFAULT;
366                         break;
367                 }

369                 if (cmd == F_O_GETLK) {
370                         /*
371                          * Return an SVR3 flock structure to the user.
372                          */
373                         obf.l_type = (int16_t)bf.l_type;
374                         obf.l_whence = (int16_t)bf.l_whence;
375                         obf.l_start = (int32_t)bf.l_start;
376                         obf.l_len = (int32_t)bf.l_len;
377                         if (bf.l_sysid > SHRT_MAX || bf.l_pid > SHRT_MAX) {
378                                 /*
379                                  * One or both values for the above fields
380                                  * is too large to store in an SVR3 flock
381                                  * structure.
382                                  */
```

```
383                                 error = EOVERFLOW;
384                                 break;
385                         }
386                         obf.l_sysid = (int16_t)bf.l_sysid;
387                         obf.l_pid = (int16_t)bf.l_pid;
388                         if (copyout(&obf, (void *)arg, sizeof (obf)))
389                                 error = EFAULT;
390                 } else if (cmd == F_GETLK) {
391                         /*
392                          * Copy out SVR4 flock.
393                          */
394                         int i;

396                         if (bf.l_start > maxoffset || bf.l_len > maxoffset) {
397                                 error = EOVERFLOW;
398                                 break;
399                         }

401                         if (datamodel == DATAMODEL_NATIVE) {
402                                 for (i = 0; i < 4; i++)
403                                         sbf.l_pad[i] = 0;
404                                 /*
405                                  * XXX  In an LP64 kernel with an LP64
406                                  *      application there's no need to do a
407                                  *      structure copy here as currently
408                                  *      struct flock == struct flock64.
409                                  *      We did it this way to avoid more
410                                  *      conditional compilation.
411                                  */
412                                 sbf.l_type = bf.l_type;
413                                 sbf.l_whence = bf.l_whence;
414                                 sbf.l_start = (off_t)bf.l_start;
415                                 sbf.l_len = (off_t)bf.l_len;
416                                 sbf.l_sysid = bf.l_sysid;
417                                 sbf.l_pid = bf.l_pid;
418                                 if (copyout(&sbf, (void *)arg, sizeof (sbf)))
419                                         error = EFAULT;
420                         }
421 #if defined(_SYSCALL32_IMPL)
422                         else {
423                                 struct flock32 sbf32;
424                                 if (bf.l_start > MAXOFF32_T ||
425                                     bf.l_len > MAXOFF32_T) {
426                                         error = EOVERFLOW;
427                                         break;
428                                 }
429                                 for (i = 0; i < 4; i++)
430                                         sbf32.l_pad[i] = 0;
431                                 sbf32.l_type = (int16_t)bf.l_type;
432                                 sbf32.l_whence = (int16_t)bf.l_whence;
433                                 sbf32.l_start = (off32_t)bf.l_start;
434                                 sbf32.l_len = (off32_t)bf.l_len;
435                                 sbf32.l_sysid = (int32_t)bf.l_sysid;
436                                 sbf32.l_pid = (pid32_t)bf.l_pid;
437                                 if (copyout(&sbf32,
438                                     (void *)arg, sizeof (sbf32)))
439                                         error = EFAULT;
440                         }
441 #endif
442                 }
443                 break;
453 /* ONC_PLUS EXTRACT END */

445         case F_CHKFL:
446                 /*
447                  * This is for internal use only, to allow the vnode layer
```



```
448                  * to validate a flags setting before applying it.  User
449                  * programs can't issue it.
450                  */
451                 error = EINVAL;
452                 break;

454         case F_ALLOCSP:
455         case F_FREESP:
456         case F_ALLOCSP64:
457         case F_FREESP64:
458                 /*
459                  * Test for not-a-regular-file (and returning EINVAL)
460                  * before testing for open-for-writing (and returning EBADF).
461                  * This is relied upon by posix_fallocate() in libc.
462                  */
463                 if (vp->v_type != VREG) {
464                         error = EINVAL;
465                         break;
466                 }

468                 if ((flag & FWRITE) == 0) {
469                         error = EBADF;
470                         break;
471                 }

473                 if (datamodel != DATAMODEL_ILP32 &&
474                     (cmd == F_ALLOCSP64 || cmd == F_FREESP64)) {
475                         error = EINVAL;
476                         break;
477                 }

479 #if defined(_ILP32) || defined(_SYSCALL32_IMPL)
480                 if (datamodel == DATAMODEL_ILP32 &&
481                     (cmd == F_ALLOCSP || cmd == F_FREESP)) {
482                         struct flock32 sbf32;
483                         /*
484                          * For compatibility we overlay an SVR3 flock on an SVR4
485                          * flock.  This works because the input field offsets
486                          * in "struct flock" were preserved.
487                          */
488                         if (copyin((void *)arg, &sbf32, sizeof (sbf32))) {
489                                 error = EFAULT;
490                                 break;
491                         } else {
492                                 bf.l_type = sbf32.l_type;
493                                 bf.l_whence = sbf32.l_whence;
494                                 bf.l_start = (off64_t)sbf32.l_start;
495                                 bf.l_len = (off64_t)sbf32.l_len;
496                                 bf.l_sysid = sbf32.l_sysid;
497                                 bf.l_pid = sbf32.l_pid;
498                         }
499                 }
500 #endif /* _ILP32 || _SYSCALL32_IMPL */

502 #if defined(_LP64)
503                 if (datamodel == DATAMODEL_LP64 &&
504                     (cmd == F_ALLOCSP || cmd == F_FREESP)) {
505                         if (copyin((void *)arg, &bf, sizeof (bf))) {
506                                 error = EFAULT;
507                                 break;
508                         }
509                 }
510 #endif /* defined(_LP64) */

512 #if !defined(_LP64) || defined(_SYSCALL32_IMPL)
513                 if (datamodel == DATAMODEL_ILP32 &&
```

```
514                     (cmd == F_ALLOCSP64 || cmd == F_FREESP64)) {
515                         if (copyin((void *)arg, &bf64_32, sizeof (bf64_32))) {
516                                 error = EFAULT;
517                                 break;
518                         } else {
519                                 /*
520                                  * Note that the size of flock64 is different in
521                                  * the ILP32 and LP64 models, due to the l_pad
522                                  * field. We do not want to assume that the
523                                  * flock64 structure is laid out the same in
524                                  * ILP32 and LP64 environments, so we will
525                                  * copy in the ILP32 version of flock64
526                                  * explicitly and copy it to the native
527                                  * flock64 structure.
528                                  */
529                                 bf.l_type = (short)bf64_32.l_type;
530                                 bf.l_whence = (short)bf64_32.l_whence;
531                                 bf.l_start = bf64_32.l_start;
532                                 bf.l_len = bf64_32.l_len;
533                                 bf.l_sysid = (int)bf64_32.l_sysid;
534                                 bf.l_pid = (pid_t)bf64_32.l_pid;
535                         }
536                 }
537 #endif /* !defined(_LP64) || defined(_SYSCALL32_IMPL) */

539                 if (cmd == F_ALLOCSP || cmd == F_FREESP)
540                         error = flock_check(vp, &bf, offset, maxoffset);
541                 else if (cmd == F_ALLOCSP64 || cmd == F_FREESP64)
542                         error = flock_check(vp, &bf, offset, MAXOFFSET_T);
543                 if (error)
544                         break;

546                 if (vp->v_type == VREG && bf.l_len == 0 &&
547                     bf.l_start > OFFSET_MAX(fp)) {
548                         error = EFBIG;
549                         break;
550                 }

552                 /*
553                  * Make sure that there are no conflicting non-blocking
554                  * mandatory locks in the region being manipulated. If
555                  * there are such locks then return EACCES.
556                  */
557                 if ((error = flock_get_start(vp, &bf, offset, &start)) != 0)
558                         break;

560                 if (nbl_need_check(vp)) {
561                         u_offset_t      begin;
562                         ssize_t         length;

564                         nbl_start_crit(vp, RW_READER);
565                         in_crit = 1;
566                         vattr.va_mask = AT_SIZE;
567                         if ((error = VOP_GETATTR(vp, &vattr, 0, CRED(), NULL))
568                             != 0)
569                                 break;
570                         begin = start > vattr.va_size ? vattr.va_size : start;
571                         length = vattr.va_size > start ? vattr.va_size - start :
572                             start - vattr.va_size;
573                         if (nbl_conflict(vp, NBL_WRITE, begin, length, 0,
574                             NULL)) {
575                                 error = EACCES;
576                                 break;
577                         }
578                 }
```



```
580                 if (cmd == F_ALLOCSP64)
581                         cmd = F_ALLOCSP;
582                 else if (cmd == F_FREESP64)
583                         cmd = F_FREESP;

585                 error = VOP_SPACE(vp, cmd, &bf, flag, offset, fp->f_cred, NULL);

587                 break;

589 #if !defined(_LP64) || defined(_SYSCALL32_IMPL)
600 /* ONC_PLUS EXTRACT START */
590         case F_GETLK64:
591         case F_SETLK64:
592         case F_SETLKW64:
593         case F_SETLK64_NBMAND:
594                 /*
595                  * Large Files: Here we set cmd as *LK and send it to
596                  * lower layers. *LK64 is only for the user land.
597                  * Most of the comments described above for F_SETLK
598                  * applies here too.
599                  * Large File support is only needed for ILP32 apps!
600                  */
601                 if (datamodel != DATAMODEL_ILP32) {
602                         error = EINVAL;
603                         break;
604                 }

606                 if (cmd == F_GETLK64)
607                         cmd = F_GETLK;
608                 else if (cmd == F_SETLK64)
609                         cmd = F_SETLK;
610                 else if (cmd == F_SETLKW64)
611                         cmd = F_SETLKW;
612                 else if (cmd == F_SETLK64_NBMAND)
613                         cmd = F_SETLK_NBMAND;

615                 /*
616                  * Note that the size of flock64 is different in the ILP32
617                  * and LP64 models, due to the sucking l_pad field.
618                  * We do not want to assume that the flock64 structure is
619                  * laid out in the same in ILP32 and LP64 environments, so
620                  * we will copy in the ILP32 version of flock64 explicitly
621                  * and copy it to the native flock64 structure.
622                  */

624                 if (copyin((void *)arg, &bf64_32, sizeof (bf64_32))) {
625                         error = EFAULT;
626                         break;
627                 }

629                 bf.l_type = (short)bf64_32.l_type;
630                 bf.l_whence = (short)bf64_32.l_whence;
631                 bf.l_start = bf64_32.l_start;
632                 bf.l_len = bf64_32.l_len;
633                 bf.l_sysid = (int)bf64_32.l_sysid;
634                 bf.l_pid = (pid_t)bf64_32.l_pid;

636                 if ((error = flock_check(vp, &bf, offset, MAXOFFSET_T)) != 0)
637                         break;

639                 if ((error = VOP_FRLOCK(vp, cmd, &bf, flag, offset,
640                     NULL, fp->f_cred, NULL)) != 0)
641                         break;

643                 if ((cmd == F_GETLK) && bf.l_type == F_UNLCK) {
644                         if (copyout(&bf.l_type, &((struct flock *)arg)->l_type,
```

```
645                             sizeof (bf.l_type)))
646                                 error = EFAULT;
647                         break;
648                 }

650                 if (cmd == F_GETLK) {
651                         int i;

653                         /*
654                          * We do not want to assume that the flock64 structure
655                          * is laid out in the same in ILP32 and LP64
656                          * environments, so we will copy out the ILP32 version
657                          * of flock64 explicitly after copying the native
658                          * flock64 structure to it.
659                          */
660                         for (i = 0; i < 4; i++)
661                                 bf64_32.l_pad[i] = 0;
662                         bf64_32.l_type = (int16_t)bf.l_type;
663                         bf64_32.l_whence = (int16_t)bf.l_whence;
664                         bf64_32.l_start = bf.l_start;
665                         bf64_32.l_len = bf.l_len;
666                         bf64_32.l_sysid = (int32_t)bf.l_sysid;
667                         bf64_32.l_pid = (pid32_t)bf.l_pid;
668                         if (copyout(&bf64_32, (void *)arg, sizeof (bf64_32)))
669                                 error = EFAULT;
670                 }
671                 break;
683 /* ONC_PLUS EXTRACT END */
672 #endif /* !defined(_LP64) || defined(_SYSCALL32_IMPL) */

686 /* ONC_PLUS EXTRACT START */
674         case F_SHARE:
675         case F_SHARE_NBMAND:
676         case F_UNSHARE:

678                 /*
679                  * Copy in input fields only.
680                  */
681                 if (copyin((void *)arg, &fsh, sizeof (fsh))) {
682                         error = EFAULT;
683                         break;
684                 }

686                 /*
687                  * Local share reservations always have this simple form
688                  */
689                 shr.s_access = fsh.f_access;
690                 shr.s_deny = fsh.f_deny;
691                 shr.s_sysid = 0;
692                 shr.s_pid = ttoproc(curthread)->p_pid;
693                 shr_own.sl_pid = shr.s_pid;
694                 shr_own.sl_id = fsh.f_id;
695                 shr.s_own_len = sizeof (shr_own);
696                 shr.s_owner = (caddr_t)&shr_own;
697                 error = VOP_SHRLOCK(vp, cmd, &shr, flag, fp->f_cred, NULL);
711 /* ONC_PLUS EXTRACT END */
698                 break;

700         default:
701                 error = EINVAL;
702                 break;
703         }

705         if (in_crit)
706                 nbl_end_crit(vp);
```



```
708 done:
709         releasef(fdes);
710 out:
711         if (error)
712                 return (set_errno(error));
713         return (retval);
714 }

730 /* ONC_PLUS EXTRACT START */
716 int
717 flock_check(vnode_t *vp, flock64_t *flp, offset_t offset, offset_t max)
718 {
719         struct vattr    vattr;
720         int     error;
721         u_offset_t start, end;

723         /*
724          * Determine the starting point of the request
725          */
726         switch (flp->l_whence) {
727         case 0:                 /* SEEK_SET */
728                 start = (u_offset_t)flp->l_start;
729                 if (start > max)
730                         return (EINVAL);
731                 break;
732         case 1:                 /* SEEK_CUR */
733                 if (flp->l_start > (max - offset))
734                         return (EOVERFLOW);
735                 start = (u_offset_t)(flp->l_start + offset);
736                 if (start > max)
737                         return (EINVAL);
738                 break;
739         case 2:                 /* SEEK_END */
740                 vattr.va_mask = AT_SIZE;
741                 if (error = VOP_GETATTR(vp, &vattr, 0, CRED(), NULL))
742                         return (error);
743                 if (flp->l_start > (max - (offset_t)vattr.va_size))
744                         return (EOVERFLOW);
745                 start = (u_offset_t)(flp->l_start + (offset_t)vattr.va_size);
746                 if (start > max)
747                         return (EINVAL);
748                 break;
749         default:
750                 return (EINVAL);
751         }

753         /*
754          * Determine the range covered by the request.
755          */
756         if (flp->l_len == 0)
757                 end = MAXEND;
758         else if ((offset_t)flp->l_len > 0) {
759                 if (flp->l_len > (max - start + 1))
760                         return (EOVERFLOW);
761                 end = (u_offset_t)(start + (flp->l_len - 1));
762                 ASSERT(end <= max);
763         } else {
764                 /*
765                  * Negative length; why do we even allow this ?
766                  * Because this allows easy specification of
767                  * the last n bytes of the file.
768                  */
769                 end = start;
770                 start += (u_offset_t)flp->l_len;
771                 (start)++;
772                 if (start > max)
```

```
773                         return (EINVAL);
774                 ASSERT(end <= max);
775         }
776         ASSERT(start <= max);
777         if (flp->l_type == F_UNLCK && flp->l_len > 0 &&
778             end == (offset_t)max) {
779                 flp->l_len = 0;
780         }
781         if (start  > end)
782                 return (EINVAL);
783         return (0);
784 }
```

unchanged_portion_omitted_

```
842 /* ONC_PLUS EXTRACT END */
```

```
**********************************************************
    5304 Thu Jul 11 01:30:12 2013
new/usr/src/uts/intel/Makefile
first pass
**********************************************************
#
# 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
#
# uts/intel/Makefile
#
# Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
#
#       This makefile drives the production of all implementation architecture
#       independent modules for Intel processors.

UTSBASE = ..

include Makefile.intel

LINT_KMODS_X1   = $(LINT_KMODS:nsmb=)
LINT_KMODS_X2   = $(LINT_KMODS_X1:smbfs=)
LINT_KMODLIBS   = $(LINT_KMODS_X2:e1000g=)
LINT_LIBS       = $(LINT_LIB) $(GEN_LINT_LIB) \
                  $(LINT_KMODLIBS:%=$(LINT_LIB_DIR)/llib-l%.ln) \
                  $(CLOSED_LINT_KMODS:%=$(LINT_LIB_DIR)/llib-l%.ln)

# EXPORT DELETE START
$(CLOSED_BUILD)LINT_LIBS        += $(SVVS_KMODS:%=$(LINT_LIB_DIR)/llib-l%.ln)
$(CLOSED_BUILD)LINT_CLOSED_XMOD4        = $(CLOSED_XMODS:bnx=)
$(CLOSED_BUILD)LINT_CLOSED_XMOD3        = $(LINT_CLOSED_XMOD4:bnxe=)
$(CLOSED_BUILD)LINT_CLOSED_XMOD2        = $(LINT_CLOSED_XMOD3:lsimega=)
$(CLOSED_BUILD)LINT_CLOSED_XMOD1        = $(LINT_CLOSED_XMOD2:adpu320=)
$(CLOSED_BUILD)LINT_LIBS        += $(LINT_XMODLIBS:%=$(LINT_LIB_DIR)/llib-l%.ln)

#
# dprov is delivered in the SUNWcryptoint package.
#
DRV_KMODS       += dprov

# EXPORT DELETE END

#
#
def             :=      TARGET= def
def.prereq      :=      TARGET= def
all             :=      TARGET= all
all.prereq      :=      TARGET= all
install         :=      TARGET= install
install.prereq  :=      TARGET= all
```



```
clean           :=      TARGET= clean
clobber         :=      TARGET= clobber
lint            :=      TARGET= lint
lint.prereq     :=      TARGET= lint
modlintlib      :=      TARGET= modlintlib
modlist         :=      TARGET= modlist
modlist         :=      NO_STATE= -K $$MODSTATE$$$$
clean.lint      :=      TARGET= clean.lint
check           :=      TARGET= check
install_h       :=      TARGET= install_h
install_h.prereq        :=      TARGET= install_h

.KEEP_STATE:

.PARALLEL:      $(PARALLEL_KMODS) $(CLOSED_KMODS) $(SVVS) $(XMODS) \
                $(CLOSED_XMODS) config $(LINT_DEPS)

def all install clean clobber modlist: $(KMODS) $(CLOSED_KMODS) \
        $(SVVS) $(XMODS) $(CLOSED_XMODS) config


#
# Privilege constants
#
# NOTE: The rules for generating priv_const.c file are shared between all
# processor architectures and and should be kept in sync. If they are changed in
# this file make sure that sparc rules are updated as well.
#
PRIVS_C = $(SRC)/uts/common/os/priv_const.c

$(PRIVS_C): $(PRIVS_AWK) $(PRIVS_DEF)
        $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) cfile=$@

#
# Prerequisites
#
# The uts/Makefile defines build parallelism for x86 platforms such that i86pc,
# i86xpv and intel are all built in parallel. This requires building certain
# parts before the parallel build can start. The uts/Makefile appends the
# '.prereq' string to the original target and executes this Makefile to build
# any prerequisites needed before the full parallel build can start. After that
# make continues with normal targets.
#
# Any build prerequisites for x86 builds should be described here.
#
# genassym is used to build intel/dtrace and genunix, so it should be built
# first.
#
# priv_const.c is required to build genunix.
#
# genunix is used by everyone to ctf-merge with. Genunix is CTF-merged with
#   intel/ip so as a side effect this dependency builds intel/ip as part of the
#   prerequisites.
#
# intel/dtrace depends on i86pc/genassym, so we need to build both
# i86pc/genassym and intel/genassym.
#
all.prereq install.prereq def.prereq: genunix FRC
        @cd ../i86pc/genassym; pwd; $(MAKE) $(@:%.prereq=%)

#
# i86pc lint libraries should be built first
#
lint.prereq: FRC
        @cd ../i86pc; pwd; $(MAKE) $(NO_STATE) lint
```

```
125 #
126 # Nothing to do for any other prerequisite targets.
127 #
128 %.prereq:

130 genunix: $(PRIVS_C)

132 modlintlib clean.lint: $(LINT_KMODS) $(CLOSED_LINT_KMODS) $(SVVS) \
133         $(XMODS) $(CLOSED_XMODS)

135 $(KMODS) $(SUBDIRS) config:     FRC
136         @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

138 $(CLOSED_KMODS):        FRC
139         cd $(CLOSED)/uts/intel/$@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

141 $(XMODS):       FRC
142         @if [ -f $@/Makefile  ]; then \
143                 cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
144         else \
145                 true; \
146         fi

148 $(SVVS) $(CLOSED_XMODS):        FRC
149         @if [ -f $(CLOSED)/uts/intel/$@/Makefile  ]; then \
150                 cd $(CLOSED)/uts/intel/$@; pwd; \
151                     $(MAKE) $(NO_STATE) $(TARGET); \
152         else \
153                 true; \
154         fi

156 install_h check:        FRC
157         @cd sys; pwd; $(MAKE) $(TARGET)
158         @cd asm; pwd; $(MAKE) $(TARGET)
159         @cd ia32/sys; pwd; $(MAKE) $(TARGET)
160         @cd amd64/sys; pwd; $(MAKE) $(TARGET)

162 #
163 # Work-around to disable acpica global crosscheck lint warnings
164 #
165 LGREP.intel =   grep -v 'intel/io/acpica'

167 #
168 #       Full kernel lint target.
169 #
170 LINT_TARGET     = globallint

172 # workaround for multiply defined errors
173 globallint := LINTFLAGS += -erroff=E_NAME_MULTIPLY_DEF2

175 globallint:
176         @pwd
177         @-$(ECHO) "\nFULL KERNEL: global crosschecks:"
178         @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.intel) | $(LGREP.2)

180 lint:   modlintlib .WAIT $(LINT_DEPS)

185 # EXPORT DELETE START

187 EXPORT_SRC:
188         $(RM) Makefile+
189         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
190             < Makefile > Makefile+
191         $(MV) Makefile+ Makefile
192         $(CHMOD) 444 Makefile
```



```
194 # EXPORT DELETE END

182 include ../Makefile.targ
```

```
**********************************************************
    3228 Thu Jul 11 01:30:13 2013
new/usr/src/uts/sparc/Makefile
first pass
**********************************************************
```

```
#
# 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
#
#
#ident  "%Z%%M% %I%     %E% SMI"
#
# Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# uts/sparc/Makefile
#
#       This makefile drives the production of all implementation architecture
#       independent modules for the SPARC processor. (For those unsure, this
#       means the module will run on all SPARC processor based machines
#       running SunOS.)

UTSBASE = ..

include Makefile.sparc

LINT_KMODS_X1   = $(LINT_KMODS:nsmb=)
LINT_KMODS_X2   = $(LINT_KMODS_X1:smbfs=)
LINT_KMODLIBS   = $(LINT_KMODS_X2:e1000g=)
LINT_LIBS       = $(LINT_LIB) $(GEN_LINT_LIB) \
                  $(LINT_KMODLIBS:%=$(LINT_LIB_DIR)/llib-l%.ln)

$(CLOSED_BUILD)LINT_LIBS += $(CLOSED_LINT_KMODS:%=$(LINT_LIB_DIR)/llib-l%.ln)

# EXPORT DELETE START
$(CLOSED_BUILD)LINT_LIBS       += $(SVVS_KMODS:%=$(LINT_LIB_DIR)/llib-l%.ln)
LINT_LIBS       += $(LINT_XMODLIBS:%=$(LINT_LIB_DIR)/llib-l%.ln)
$(CLOSED_BUILD)LINT_LIBS += $(CLOSED_XMODS:%=$(LINT_LIB_DIR)/llib-l%.ln)

DRV_KMODS       += dprov

# EXPORT DELETE END

def             :=      TARGET= def
all             :=      TARGET= all
install         :=      TARGET= install
clean           :=      TARGET= clean
clobber         :=      TARGET= clobber
lint            :=      TARGET= lint
modlintlib      :=      TARGET= modlintlib
```



```
modlist         :=      TARGET= modlist
modlist         :=      NO_STATE= -K $$MODSTATE$$$$
clean.lint      :=      TARGET= clean.lint
check           :=      TARGET= check
install_h       :=      TARGET= install_h

.KEEP_STATE:

.PARALLEL:      $(PARALLEL_KMODS) $(CLOSED_KMODS) $(SVVS) $(XMODS) \
                $(CLOSED_XMODS) config $(LINT_DEPS)

def all install clean clobber modlist: $(KMODS) $(CLOSED_KMODS) $(SVVS) \
        $(XMODS) $(CLOSED_XMODS) config

modlintlib clean.lint: $(LINT_KMODS) $(CLOSED_LINT_KMODS) $(SVVS) \
        $(XMODS) $(CLOSED_XMODS)

$(KMODS) config:        FRC
        @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

$(CLOSED_KMODS):        FRC
        cd $(CLOSED)/uts/sparc/$@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

$(XMODS):       FRC
        @if [ -f $@/Makefile  ]; then \
                cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
        else \
                true; \
        fi

$(SVVS) $(CLOSED_XMODS):        FRC
        @if [ -f $(CLOSED)/uts/sparc/$@/Makefile  ]; then \
                cd $(CLOSED)/uts/sparc/$@; pwd; \
                    $(MAKE) $(NO_STATE) $(TARGET); \
        else \
                true; \
        fi

install_h check:        FRC
        @cd asm; pwd; $(MAKE) $(TARGET)
        @cd sys; pwd; $(MAKE) $(TARGET)
        @cd v7/sys; pwd; $(MAKE) $(TARGET)
        @cd v9/sys; pwd; $(MAKE) $(TARGET)

#
#       Full kernel lint target.
#
LINT_TARGET     = globallint

globallint:
        @-$(ECHO) "\nFULL KERNEL: global crosschecks:"
        @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.2)

lint:   modlintlib .WAIT $(LINT_DEPS)

# EXPORT DELETE START

EXPORT_SRC:
        $(RM) Makefile+
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
            < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile

# EXPORT DELETE END
```

```
 114 include ../Makefile.targ
```

```
**********************************************************
    8755 Thu Jul 11 01:30:13 2013
new/usr/src/uts/sun4u/Makefile
first pass
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#       This makefile drives the production of all implementation architecture
#       dependent modules for the sun4u architecture.
#

UTSBASE = ..

include Makefile.sun4u

#
#       The following are SPARC specific (rather than sun4u) specific modules
#       which are required for the sun4u kernel to completely lint. They are
#       not involved in the build in any other way. In order to minimize
#       build time, it is assumed that they are up to date. But since sun4u
#       is really a separate architecture we cannot use the v7 sparc modules.
#
SPARC_LIB_DIR    = $(UTSBASE)/sparc/lint-libs/$(OBJS_DIR)

SPARC_LINTS      =

#
#
#
LINT_LIBS        = $(LINT_LIB) \
                   $(LINT_KMODS:%=$(LINT_LIB_DIR)/llib-l%.ln) \
                   $(CLOSED_LINT_KMODS:%=$(LINT_LIB_DIR)/llib-l%.ln) \
                   $(SPARC_LINTS:%=$(SPARC_LIB_DIR)/llib-l%.ln)


def             :=      TARGET= def
def.prereq      :=      TARGET= def
all             :=      TARGET= all
all.prereq      :=      TARGET= all
install         :=      TARGET= install
install.prereq  :=      TARGET= all
install_h       :=      TARGET= install_h
install_h.prere :=      TARGET= install_h
clean           :=      TARGET= clean
```



```
clobber         :=      TARGET= clobber
lint            :=      TARGET= lint
lint.prereq     :=      TARGET= lint
lintlib         :=      TARGET= lintlib
modlintlib      :=      TARGET= modlintlib
modlist         :=      TARGET= modlist
modlist modlist.sparc :=        NO_STATE= -K $$MODSTATE$$$$
clean.lint      :=      TARGET= clean.lint
check           :=      TARGET= check

.KEEP_STATE:

.PARALLEL:      $(PARALLEL_KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
                $(IMPLEMENTATIONS) $(CLOSED_IMPLEMENTATIONS) \
                modlist modlist.sparc

# Override for CPU_KMODS... they cannot be built
# in parallel
.NO_PARALLEL:   $(CPU_KMODS)

def all clean clobber clean.lint: genassym unix .WAIT \
        $(KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
        $(IMPLEMENTATIONS) $(CLOSED_IMPLEMENTATIONS)

# list the modules under sun4u.
modlist: unix $(KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
        $(IMPLEMENTATIONS:.WAIT=) $(CLOSED_IMPLEMENTATIONS)

# list the modules for Install -k sun4u.
modlist.karch: modlist modlist.sparc

modlist.sparc:
        @cd $(SRC)/uts/sparc; pwd; $(MAKE) $(NO_STATE) modlist

install: install_platforms genassym unix .WAIT $(KMODS) $(CLOSED_KMODS) \
        $(XMODS) $(CLOSED_XMODS) $(IMPLEMENTATIONS) $(CLOSED_IMPLEMENTATIONS)

lintlib:        unix

modlintlib:     $(LINT_KMODS) $(CLOSED_LINT_KMODS)

genassym unix $(KMODS): FRC
        @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

#
# Privilege constants
#
# NOTE: The rules for generating priv_const.c file are shared between all
# processor architectures and should be kept in sync. If they are changed in
# this file make sure that x86 rules are updated as well.
#
PRIVS_C = $(UTSBASE)/common/os/priv_const.c

$(PRIVS_C): $(PRIVS_AWK) $(PRIVS_DEF)
        $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) cfile=$@


#
# Prerequisites
#
# The uts/Makefile defines build parallelism for sun4 platforms such that sparc,
# sun4u and sun4v are all built in parallel. Also this Makefile specifies that
# all IMPLEMENTATIONS sun4u sub-platforms are built in parallel. This requires
# building certain parts before the parallel build can start. The uts/Makefile
# appends the '.prereq' string to the original target and executes this Makefile
# to build any prerequisites needed before the full parallel build can start.
```

```
128 # After that make continues with normal targets.
129 #
130 # Any build prerequisites for sun4 and IMPLEMENTATIONS builds should be
131 # described here.
132 #
133 # genassym is used to build dtrace and genunix, so it should be built first.
134 #
135 # priv_const.c is required to build genunix.
136 #
137 # genunix is used by everyone to ctfmerge with. Genunix is merged with sparc/ip
138 #   so as a side effect this dependency builds sparc/ip as part of the
139 #   prerequisites.
140 #
141 # unix is not required by itself but several sun4u platforms require
142 #   sun4u/platmod to be present. The easiest way to achieve this is to build
143 #   sun4u/unix first since sun4u/unix Makefile builds sun4u/platform correctly.
144 #   This causes full sun4u/unix to be built before all sun4u platforms and
145 #   before uts/sun4v and uts/sparc, but it acceptable since it is not spending
146 #   too much time building sun4u/unix.
147 #
148 all.prereq def.prereq install.prereq: genassym genunix unix

150 #
151 # Various sun4u platforms expect proto/root_sparc/platform/sun4u/include to be
152 # present. This is handled by running make install_h in sun4u/unix directory
153 # first.
154 #
155 install_h.prereq: FRC
156         @cd sys; pwd; $(MAKE) $(TARGET)

158 #
159 # sun4u/unix and sun4u/genunix should be linted first since sparc does global
160 # cross-check with these lint libraries. The sun4u/unix and sun4u/genunix can be
161 # linted in parallel.
162 #
163 LINT_PREREQ = unix.lint genunix.lint
164 lint.prereq: $(LINT_PREREQ)

166 .PARALLEL:  $(LINT_PREREQ)

168 $(LINT_PREREQ):
169         @cd $(@:%.lint=%); pwd; $(MAKE) $(TARGET)

171 #
172 # Nothing to do with any other prerequisites
173 #
174 %.prereq:

176 #
177 # Platform inter-dependencies
178 #
179 lw8: serengeti

181 quasar: darwin

183 #
184 # The genunix requires priv_const.c file to be generated first.
185 #
186 genunix: $(PRIVS_C)

188 #
189 # Rules
190 #

192 $(IMPLEMENTATIONS):     FRC
193         @cd $@; pwd; THISIMPL=$@ $(MAKE) $(NO_STATE) $(TARGET)
```



```
195 $(CLOSED_IMPLEMENTATIONS):      FRC
196         cd $(CLOSED)/uts/sun4u/$@; pwd; \
197             THISIMPL=$@ $(MAKE) $(NO_STATE) $(TARGET); \

199 $(XMODS):       FRC
200         @if [ -f $@/Makefile  ]; then \
201                 cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
202         else \
203                 true; \
204         fi

206 $(CLOSED_XMODS):        FRC
207         @if [ -f $(CLOSED)/uts/sun4u/$@/Makefile ]; then \
208                 cd $(CLOSED)/uts/sun4u/$@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
209         else \
210                 true; \
211         fi

213 $(CLOSED_KMODS):        FRC
214         cd $(CLOSED)/uts/sun4u/$@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

216 install_h check: install_platforms $(IMPLEMENTATIONS) \
217         $(CLOSED_IMPLEMENTATIONS) FRC
218         @cd sys; pwd; $(MAKE) $(TARGET)
219         @cd vm; pwd; $(MAKE) $(TARGET)

221 #
222 # Rules for the /platforms directories.  This is hardwired here because
223 # the first stage of the project (KBI) only implements the userland
224 # changes, but the only reasonable place to record the aliases is
225 # here in kernel land.
226 #
227 $(ROOT_PLAT_DIRS): $(ROOT_PLAT_DIR)
228         -$(INS.dir)

230 #
231 # create directories in /usr/platform/ for the implementations that are
232 # defined in $(IMPLEMENTED_PLATFORM)
233 #       (eg. SUNW,Ultra-1)
234 #
235 # Foreach $(IMPLEMENTED_PLATFORM) there can be a list of $(LINKED_PLATFORMS)
236 # that are linked to it.
237 #
238 $(USR_PLAT_DIR)/$(IMPLEMENTED_PLATFORM):        $(USR_PLAT_DIR)
239         -$(INS.dir)

241 #
242 # create the links in /usr/platform/ foreach $(LINKED_PLATFORMS)
243 # to it's corresponding $(IMPLEMENTED_PLATFORM).
244 #
245 PLATFORMS       = $(LINKED_PLATFORMS)

247 $(USR_PLAT_DIRS): $(USR_PLAT_DIR)
248         $(INS.slink3)

250 PLATFORMS       += $(IMPLEMENTED_PLATFORM)

252 #
253 # Make the /platforms directories.  This is hardwired here because
254 # the first stage of the project (KBI) only implements the userland
255 # changes, but the only reasonable place to record the aliases is
256 # here in kernel land.
257 #
258 install_platforms:      $(ROOT_PSM_DIR) $(USR_PSM_DIR) \
259                         $(ROOT_PLAT_DIRS) $(USR_PLAT_DIRS) \
```

```
260                         $(USR_DESKTOP_DIR) $(USR_DESKTOP_INC_DIR) \
261                         $(USR_DESKTOP_SBIN_DIR) $(USR_DESKTOP_LIB_DIR)

263 #
264 # rules for making include, sbin, lib dirs/links in
265 # /usr/platform/$(PLATFORM)/ for desktop platforms
266 #
267 $(USR_DESKTOP_INC_DIR):         $(USR_DESKTOP_DIR)
268         $(INS.slink4)

270 $(USR_DESKTOP_SBIN_DIR):        $(USR_DESKTOP_DIR)
271         $(INS.slink5)

273 $(USR_DESKTOP_LIB_DIR):         $(USR_DESKTOP_DIR)
274         -$(INS.dir)

276 #
277 #       Full kernel lint target.
278 #
279 LINT_TARGET     = globallint

281 globallint:
282         @pwd
283         @-$(ECHO) "\nSUN4U KERNEL: global crosschecks:"
284         @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.2)

286 lint:   lintlib .WAIT modlintlib .WAIT $(SPARC_LINTS) $(LINT_DEPS) \
287         $(IMPLEMENTATIONS) $(CLOSED_IMPLEMENTATIONS) $(CPU_KMODS)

289 # EXPORT DELETE START

291 EXPORT_SRC:
292         $(RM) Makefile+
293         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
294             < Makefile > Makefile+
295         $(MV) Makefile+ Makefile
296         $(CHMOD) 444 Makefile

298 # EXPORT DELETE END

289 include ../Makefile.targ

291 #
292 # Cross-reference customization: build a cross-reference over all of the
293 # sun4u-related directories.
294 #
295 SHARED_XRDIRS   = ../sun4u ../sun4 ../sfmmu ../sparc ../sun ../common
296 CLOSED_XRDIRS   = $(SHARED_XRDIRS:../%=../% ../../../closed/uts/%)
297 XRDIRS          = $(SHARED_XRDIRS)
298 $(CLOSED_BUILD)XRDIRS   = $(CLOSED_XRDIRS:../../../closed/uts/sfmmu=)

300 XRPRUNE = i86pc

302 cscope.out tags: FRC
303         $(XREF) -x $@
```

```
**********************************************************
    8114 Thu Jul 11 01:30:14 2013
new/usr/src/uts/sun4v/Makefile
first pass
**********************************************************
   1  #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #
  25 #       This makefile drives the production of all implementation architecture
  26 #       dependent modules for the sun4v architecture.
  27 #

  29 UTSBASE = ..

  31 include Makefile.sun4v
  32 include Makefile.stpaul
  33 include Makefile.huron
  34 include Makefile.maramba
  35 include Makefile.thunder
  36 include Makefile.turgo
  37 include Makefile.congo
  38 include Makefile.monza

  40 USR_GLENDALE_DIR        = $(USR_PLAT_DIR)/SUNW,Sun-Blade-T6320
  41 USR_GLENDALE_SBIN_DIR   = $(USR_GLENDALE_DIR)/sbin
  42 USR_GLENDALE_LIB_DIR    = $(USR_GLENDALE_DIR)/lib


  45 #
  46 #       The following are SPARC specific (rather than sun4v) specific modules
  47 #       which are required for the sun4v kernel to completely lint. They are
  48 #       not involved in the build in any other way. In order to minimize
  49 #       build time, it is assumed that they are up to date. But since sun4v
  50 #       is really a separate architecture we cannot use the v7 sparc modules.
  51 #
  52 SPARC_LIB_DIR    = $(UTSBASE)/sparc/lint-libs/$(OBJS_DIR)

  54 SPARC_LINTS      =

  56 #
  57 #
  58 #
  59 LINT_LIBS        = $(LINT_LIB) \
  60                    $(LINT_KMODS:%=$(LINT_LIB_DIR)/llib-l%.ln) \
  61                    $(CLOSED_LINT_KMODS:%=$(LINT_LIB_DIR)/llib-l%.ln) \
```



```
  62                    $(SPARC_LINTS:%=$(SPARC_LIB_DIR)/llib-l%.ln)

  64 def             :=      TARGET= def
  65 all             :=      TARGET= all
  66 install         :=      TARGET= install
  67 install_h       :=      TARGET= install_h
  68 clean           :=      TARGET= clean
  69 clobber         :=      TARGET= clobber
  70 lint            :=      TARGET= lint
  71 lintlib         :=      TARGET= lintlib
  72 modlintlib      :=      TARGET= modlintlib
  73 modlist         :=      TARGET= modlist
  74 modlist modlist.sparc :=        NO_STATE= -K $$MODSTATE$$$$
  75 clean.lint      :=      TARGET= clean.lint
  76 check           :=      TARGET= check

  78 .KEEP_STATE:

  80 .PARALLEL:      $(PARALLEL_KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
  81                 modlist modlist.sparc

  83 # Override for CPU_KMODS... they cannot be built
  84 # in parallel
  85 .NO_PARALLEL:   $(CPU_KMODS)

  87 def all clean clobber clean.lint: genassym unix .WAIT \
  88         $(KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) $(IMPLEMENTATIONS)

  90 # list the modules under sun4v.
  91 modlist: unix $(KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
  92         $(IMPLEMENTATIONS)

  94 # list the modules for Install -k sun4v.
  95 modlist.karch: modlist modlist.sparc

  97 modlist.sparc:
  98         @cd $(SRC)/uts/sparc; pwd; $(MAKE) $(NO_STATE) modlist

 100 install: install_platforms genassym unix .WAIT $(KMODS) $(CLOSED_KMODS) \
 101         $(XMODS) $(CLOSED_XMODS) $(IMPLEMENTATIONS)

 103 lintlib:        unix

 105 modlintlib:     $(LINT_KMODS) $(CLOSED_LINT_KMODS)

 107 genassym unix $(KMODS): FRC
 108         @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

 110 $(IMPLEMENTATIONS):     FRC
 111         @cd $@; pwd; THISIMPL=$@ $(MAKE) $(NO_STATE) $(TARGET)

 113 $(XMODS):       FRC
 114         @if [ -f $@/Makefile  ]; then \
 115                 cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
 116         else \
 117                 true; \
 118         fi

 120 $(CLOSED_XMODS):        FRC
 121         @if [ -f $(CLOSED)/uts/sun4v/$@/Makefile  ]; then \
 122                 cd $(CLOSED)/uts/sun4v/$@; pwd; \
 123                     $(MAKE) $(NO_STATE) $(TARGET); \
 124         else \
 125                 true; \
 126         fi
```

```
$(CLOSED_KMODS):        FRC
        cd $(CLOSED)/uts/sun4v/$@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

install_h check:        install_platforms $(IMPLEMENTATIONS) FRC
        @cd sys; pwd; $(MAKE) $(TARGET)
        @cd vm; pwd; $(MAKE) $(TARGET)

#
# Rules for the /platforms directories.  This is hardwired here because
# the first stage of the project (KBI) only implements the userland
# changes, but the only reasonable place to record the aliases is
# here in kernel land.
#
$(ROOT_PLAT_DIRS): $(ROOT_PLAT_DIR)
        -$(INS.dir)

$(LINKED_PLATFORMS:%=$(ROOT_PLAT_DIR)/%): $(ROOT_PLAT_DIR)
        $(INS.slink1)

#
# create directories in /usr/platform/ for the implementations that are
# defined in $(IMPLEMENTED_PLATFORM)
#

# Foreach $(IMPLEMENTED_PLATFORM) there can be a list of $(LINKED_PLATFORMS)
# that are linked to it.
#
$(USR_PLAT_DIR)/$(IMPLEMENTED_PLATFORM):        $(USR_PLAT_DIR)
        -$(INS.dir)

#
# create the links in /usr/platform/ foreach $(LINKED_PLATFORMS)
# to it's corresponding $(IMPLEMENTED_PLATFORM).
#
PLATFORMS       = $(LINKED_PLATFORMS)

$(USR_PLAT_DIRS): $(USR_PLAT_DIR)
        $(INS.slink3)

PLATFORMS       += $(IMPLEMENTED_PLATFORM)


#
# Make the /platforms directories.  This is hardwired here because
# the first stage of the project (KBI) only implements the userland
# changes, but the only reasonable place to record the aliases is
# here in kernel land.
#

install_platforms:      $(ROOT_PSM_DIR) $(USR_PSM_DIR) \
                        $(ROOT_PLAT_DIRS) $(USR_PLAT_DIRS) \
                        $(LINKED_PLATFORMS:%=$(ROOT_PLAT_DIR)/%) \
                        $(USR_DESKTOP_DIR) $(USR_DESKTOP_INC_DIR) \
                        $(USR_DESKTOP_SBIN_DIR) $(USR_DESKTOP_LIB_DIR) \
                        $(USR_STPAUL_DIR) $(USR_STPAUL_SBIN_DIR) \
                        $(USR_STPAUL_LIB_DIR) \
                        $(USR_GLENDALE_DIR) $(USR_GLENDALE_SBIN_DIR) \
                        $(USR_GLENDALE_LIB_DIR) \
                        $(USR_HURON_DIR) \
                        $(USR_HURON_SBIN_DIR) $(USR_HURON_LIB_DIR) \
                        $(USR_MARAMBA_DIR) $(USR_MARAMBA_SBIN_DIR) \
                        $(USR_MARAMBA_LIB_DIR) \
                        $(USR_THUNDER_DIR) $(USR_THUNDER_SBIN_DIR) \
                        $(USR_THUNDER_LIB_DIR) \
                        $(USR_TURGO_DIR) $(USR_TURGO_SBIN_DIR) \
                        $(USR_TURGO_LIB_DIR) \
```



```
                        $(USR_CONGO_DIR) $(USR_CONGO_SBIN_DIR) \
                        $(USR_CONGO_LIB_DIR) \
                        $(USR_MONZA_DIR) \
                        $(USR_MONZA_SBIN_DIR) $(USR_MONZA_SBIN_LINKS)


#
# rules for making include, sbin, lib dirs/links in
# /usr/platform/$(PLATFORM)/ for desktop platforms
#
$(USR_DESKTOP_INC_DIR):         $(USR_DESKTOP_DIR)
        $(INS.slink4)

$(USR_DESKTOP_SBIN_DIR):        $(USR_DESKTOP_DIR)
        $(INS.slink5)

$(USR_DESKTOP_LIB_DIR):         $(USR_DESKTOP_DIR)
        -$(INS.dir)

$(USR_STPAUL_DIR):              $(USR_SUN4V_PLAT_DIR)
        -$(INS.dir)

$(USR_STPAUL_SBIN_DIR):         $(USR_STPAUL_DIR)
        $(INS.slink5)

$(USR_STPAUL_LIB_DIR):          $(USR_STPAUL_DIR)
        -$(INS.dir)

$(USR_HURON_DIR):               $(USR_SUN4V_PLAT_DIR)
        -$(INS.dir)

$(USR_HURON_SBIN_DIR):          $(USR_HURON_DIR)
        $(INS.slink5)

$(USR_HURON_LIB_DIR):           $(USR_HURON_DIR)
        -$(INS.dir)

$(USR_GLENDALE_DIR):            $(USR_SUN4V_PLAT_DIR)
        -$(INS.dir)

$(USR_GLENDALE_SBIN_DIR):               $(USR_GLENDALE_DIR)
        $(INS.slink5)

$(USR_GLENDALE_LIB_DIR):                $(USR_GLENDALE_DIR)
        -$(INS.dir)

$(USR_MARAMBA_DIR):             $(USR_SUN4V_PLAT_DIR)
        -$(INS.dir)

$(USR_MARAMBA_SBIN_DIR):        $(USR_MARAMBA_DIR)
        $(INS.slink5)

$(USR_MARAMBA_LIB_DIR):         $(USR_MARAMBA_DIR)
        -$(INS.dir)

$(USR_THUNDER_DIR):             $(USR_SUN4V_PLAT_DIR)
        -$(INS.dir)

$(USR_THUNDER_SBIN_DIR):        $(USR_THUNDER_DIR)
        $(INS.slink5)

$(USR_THUNDER_LIB_DIR):         $(USR_THUNDER_DIR)
        -$(INS.dir)

$(USR_TURGO_DIR):               $(USR_SUN4V_PLAT_DIR)
        -$(INS.dir)
```

```
261 $(USR_TURGO_SBIN_DIR):          $(USR_TURGO_DIR)
262         $(INS.slink5)

264 $(USR_TURGO_LIB_DIR):           $(USR_TURGO_DIR)
265         -$(INS.dir)

267 $(USR_CONGO_DIR):               $(USR_SUN4V_PLAT_DIR)
268         -$(INS.dir)

270 $(USR_CONGO_SBIN_DIR):          $(USR_CONGO_DIR)
271         $(INS.slink5)

273 $(USR_CONGO_LIB_DIR):           $(USR_CONGO_DIR)
274         -$(INS.dir)

276 $(USR_MONZA_DIR):               $(USR_SUN4V_PLAT_DIR)
277         -$(INS.dir)

279 $(USR_MONZA_SBIN_DIR):          $(USR_MONZA_DIR)
280         -$(INS.dir)

282 $(USR_MONZA_SBIN_LINKS):        $(USR_MONZA_SBIN_DIR)
283         $(INS.slink7)

285 #
286 #       Full kernel lint target.
287 #
288 LINT_TARGET     = globallint

290 globallint:
291         @-$(ECHO) "\nSUN4V KERNEL: global crosschecks:"
292         @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.2)

294 lint:   lintlib .WAIT modlintlib .WAIT $(SPARC_LINTS) $(LINT_DEPS) \
295         $(IMPLEMENTATIONS) $(LINT_CPU_KMODS)

297 # EXPORT DELETE START

299 EXPORT_SRC:
300         $(RM) Makefile+
301         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
302             < Makefile > Makefile+
303         $(MV) Makefile+ Makefile
304         $(CHMOD) 444 Makefile

306 # EXPORT DELETE END

297 include ../Makefile.targ

299 #
300 # Cross-reference customization: build a cross-reference over all of the
301 # sun4v-related directories.
302 #
303 SHARED_XRDIRS   = ../sun4v ../sun4 ../sfmmu ../sparc ../sun ../common
304 CLOSED_XRDIRS   = $(SHARED_XRDIRS:../%=../% ../../../closed/uts/%)
305 XRDIRS          = $(SHARED_XRDIRS)
306 $(CLOSED_BUILD)XRDIRS   = $(CLOSED_XRDIRS:../../../closed/uts/sfmmu=)

308 XRPRUNE = i86pc sun4u intel

310 cscope.out tags: FRC
311         $(XREF) -x $@
```

```
**********************************************************
    2016 Thu Jul 11 01:30:14 2013
new/usr/src/uts/sun4v/huron/Makefile
first pass
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 #
  22 # uts/sun4v/huron/Makefile
  23 # Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  24 # Use is subject to license terms.
  25 #
  26 #
  27 #       This makefile drives the production of the sun4v huron platform
  28 #       modules.
  29 #
  30 #       sun4v huron implementation architecture dependent
  31 #

  33 #
  34 #       Path to the base of the uts directory tree (usually /usr/src/uts).
  35 #
  36 UTSBASE = ../..

  38 #
  39 #       Include common rules.
  40 #
  41 include $(UTSBASE)/sun4v/Makefile.sun4v

  43 USR_PLAT_DIR = $(ROOT)/usr/platform

  45 include $(UTSBASE)/sun4v/Makefile.huron

  47 def             :=      TARGET= def
  48 all             :=      TARGET= all
  49 install         :=      TARGET= install
  50 install_h       :=      TARGET= install_h
  51 clean           :=      TARGET= clean
  52 clobber         :=      TARGET= clobber
  53 lint            :=      TARGET= lint
  54 lintlib         :=      TARGET= lintlib
  55 modlintlib      :=      TARGET= modlintlib
  56 modlist         :=      TARGET= modlist
  57 modlist         :=      NO_STATE= -K $$MODSTATE$$$$
  58 clean.lint      :=      TARGET= clean.lint
  59 check           :=      TARGET= check

  61 #
```



```
  62 #       Default build targets.
  63 #
  64 .KEEP_STATE:

  66 def:

  68 lintlib:      unix

  70 IMPLEMENTED_PLATFORM    = SUNW,SPARC-Enterprise-T5120
  71 LINKED_PLATFORMS        = SUNW,SPARC-Enterprise-T5220


  74 $(LINKED_PLATFORMS:%=$(USR_PLAT_DIR)/%): $(USR_PLAT_DIR)
  75         $(INS.slink3)

  77 all:

  79 install:        $(LINKED_PLATFORMS:%=$(USR_PLAT_DIR)/%)

  81 install_h check:

  83 lint:

  85 clean:

  87 clobber:        clean

  89 modlist:

  91 EXPORT_SRC:
  92         $(RM) Makefile+
  93         sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
  94                 < Makefile > Makefile+
  95         $(MV) Makefile+ Makefile
  96         $(CHMOD) 444 Makefile
  97 # EXPORT DELETE END

  99 #
```

```
**********************************************************
    2033 Thu Jul 11 01:30:15 2013
new/usr/src/uts/sun4v/maramba/Makefile
first pass
**********************************************************
```

```
#
# 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
#
#
# uts/sun4v/maramba/Makefile
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident  "%Z%%M% %I%     %E% SMI"
#
#       This makefile creates the links that point at
#       $(USR_PLAT_DIR)/SUNW,T5140
#


#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..

#
#       Include common rules.
#

include $(UTSBASE)/sun4v/Makefile.sun4v

USR_PLAT_DIR = $(ROOT)/usr/platform

include $(UTSBASE)/sun4v/Makefile.maramba

def             :=      TARGET= def
all             :=      TARGET= all
install         :=      TARGET= install
install_h       :=      TARGET= install_h
clean           :=      TARGET= clean
clobber         :=      TARGET= clobber
lint            :=      TARGET= lint
lintlib         :=      TARGET= lintlib
modlintlib      :=      TARGET= modlintlib
modlist         :=      TARGET= modlist
modlist         :=      NO_STATE= -K $$MODSTATE$$$$
clean.lint      :=      TARGET= clean.lint
check           :=      TARGET= check

#
```



```
#       Default build targets.
#
.KEEP_STATE:

modlist:

def:

lintlib:      unix

IMPLEMENTED_PLATFORM    = SUNW,T5140
LINKED_PLATFORMS        = SUNW,T5240
LINKED_PLATFORMS        += SUNW,T5440
LINKED_PLATFORMS        += SUNW,Sun-Blade-T6340

$(LINKED_PLATFORMS:%=$(USR_PLAT_DIR)/%): $(USR_PLAT_DIR)
        $(INS.slink3)

all:

install:        $(LINKED_PLATFORMS:%=$(USR_PLAT_DIR)/%)

install_h check:

lint:

clean:

clobber: clean


EXPORT_SRC:
        $(RM) Makefile+
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile
# EXPORT DELETE END

#
```

```
**********************************************************
    2570 Thu Jul 11 01:30:16 2013
new/usr/src/uts/sun4v/montoya/Makefile
first pass
**********************************************************
#
# 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
#
#
# uts/sun4v/montoya/Makefile
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident  "%Z%%M% %I%     %E% SMI"
#
#       This makefile drives the production of the sun4v montoya platform
#       modules.
#
#       sun4v montoya implementation architecture dependent
#


#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..

#
#       Include common rules.
#
include $(UTSBASE)/sun4v/montoya/Makefile.montoya

def             :=      TARGET= def
all             :=      TARGET= all
install         :=      TARGET= install
install_h       :=      TARGET= install_h
clean           :=      TARGET= clean
clobber         :=      TARGET= clobber
lint            :=      TARGET= lint
lintlib         :=      TARGET= lintlib
modlintlib      :=      TARGET= modlintlib
modlist         :=      TARGET= modlist
modlist         :=      NO_STATE= -K $$MODSTATE$$$$
clean.lint      :=      TARGET= clean.lint
check           :=      TARGET= check

#
#       Default build targets.
#
.KEEP_STATE:
```



```
def all clean clobber clean.lint modlist:       $(MONTOYA_KMODS)

lintlib:      unix

modlintlib:   $(MONTOYA_KMODS)

IMPLEMENTED_PLATFORM    = SUNW,Netra-CP3060

install:        $(ROOT_MONTOYA_DIR) $(USR_MONTOYA_DIR) \
                $(USR_MONTOYA_LIB_DIR) \
                $(ROOT_MONTOYA_LIB_DIR) \
                $(USR_MONTOYA_SBIN_DIR) \
                $(USR_MONTOYA_SBIN_LINKS) \
                .WAIT $(MONTOYA_KMODS)

$(MONTOYA_KMODS):       FRC
        @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

install_h check:        FRC

lint:   modlintlib .WAIT $(LINT_DEPS)

LINT_LIBS       = $(LINT_LIB) \
                        -L$(MONTOYA_LINT_LIB_DIR) \
                        -L$(LINT_LIB_DIR) $(LINT_KMODS:%=-l%) \
                            $(CLOSED_LINT_KMODS:%=-l%) \
                        -L$(SPARC_LIB_DIR) $(SPARC_LINTS:%=-l%)

lint.platmod:   modlintlib
        @-$(ECHO) "\n$(IMPLEMENTED_PLATFORM) platform-dependent module: global c
        @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.2)

EXPORT_SRC:
        $(RM) Makefile+
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile
# EXPORT DELETE END

#
#       Include common targets.
#
include $(UTSBASE)/$(PLATFORM)/montoya/Makefile.targ
```

```
**********************************************************
    2892 Thu Jul 11 01:30:16 2013
new/usr/src/uts/sun4v/ontario/Makefile
first pass
**********************************************************
#
# 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
#
#
# uts/sun4v/ontario/Makefile
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident  "%Z%%M% %I%     %E% SMI"
#
#       This makefile drives the production of the sun4v ontario platform
#       modules.
#
#       sun4v ontario implementation architecture dependent
#


#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..

#
#       Include common rules.
#
include $(UTSBASE)/sun4v/ontario/Makefile.ontario

def             :=      TARGET= def
all             :=      TARGET= all
install         :=      TARGET= install
install_h       :=      TARGET= install_h
clean           :=      TARGET= clean
clobber         :=      TARGET= clobber
lint            :=      TARGET= lint
lintlib         :=      TARGET= lintlib
modlintlib      :=      TARGET= modlintlib
modlist         :=      TARGET= modlist
modlist         :=      NO_STATE= -K $$MODSTATE$$$$
clean.lint      :=      TARGET= clean.lint
check           :=      TARGET= check

#
#       Default build targets.
#
.KEEP_STATE:
```



```
def all clean clobber clean.lint modlist:       $(ONTARIO_KMODS)

lintlib:      unix

modlintlib:   $(ONTARIO_KMODS)

IMPLEMENTED_PLATFORM    = SUNW,Sun-Fire-T200
LINKED_PLATFORMS        = SUNW,Sun-Fire-T1000
LINKED_PLATFORMS        += SUNW,SPARC-Enterprise-T1000
LINKED_PLATFORMS        += SUNW,Netra-T2000
LINKED_PLATFORMS        += SUNW,SPARC-Enterprise-T2000
PPLINKED_PLATFORMS      = SUNW,Netra-T2000
PPLINKED_PLATFORMS      += SUNW,SPARC-Enterprise-T2000

install:        $(ROOT_ONTARIO_DIR) $(USR_ONTARIO_DIR) \
                $(USR_ONTARIO_SBIN_DIR) \
                $(USR_ONTARIO_LIB_DIR) \
                $(ROOT_ONTARIO_LIB_DIR) \
                $(LINKED_PLATFORMS:%=$(USR_PLAT_DIR)/%) \
                $(PPLINKED_PLATFORMS:%=$(ROOT_PLAT_DIR)/%) \
                .WAIT $(ONTARIO_KMODS)

$(ONTARIO_KMODS):       FRC
        @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

install_h check:        FRC

lint:   modlintlib .WAIT $(LINT_DEPS)

LINT_LIBS       = $(LINT_LIB) \
                        -L$(ONTARIO_LINT_LIB_DIR) \
                        -L$(LINT_LIB_DIR) $(LINT_KMODS:%=-l%) \
                            $(CLOSED_LINT_KMODS:%=-l%) \
                        -L$(SPARC_LIB_DIR) $(SPARC_LINTS:%=-l%)

lint.platmod:   modlintlib
        @-$(ECHO) "\n$(IMPLEMENTED_PLATFORM) platform-dependent module: global c
        @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.2)

EXPORT_SRC:
        $(RM) Makefile+
        sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
                < Makefile > Makefile+
        $(MV) Makefile+ Makefile
        $(CHMOD) 444 Makefile
# EXPORT DELETE END

#
#       Include common targets.
#
include $(UTSBASE)/$(PLATFORM)/ontario/Makefile.targ
```