

new/exception\_lists/check\_rtime

1

```

*****
9813 Thu Feb 13 22:45:13 2014
new/exception_lists/check_rtime
Add exceptions for objects in which the omnios build system includes bad
runpaths
*****
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 #
23 # Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 #
26 # This file provides exceptions to the usual rules applied to ELF objects by
27 # check_rtime. All strings are Perl regular expressions that are compared to
28 # file paths. In addition to the standard Perl syntax, there is one extension:
29 #
30 #     MACH(dir)
31 #
32 # is expanded into a regular expression that matches the given
33 # directory, or a 64-bit subdirectory of the directory with the
34 # name of a 64-bit architecture. For example, MACH(lib) will match
35 # any of the following:
36 #
37 #     lib
38 #     lib/amd64
39 #     lib/sparcv9
40 #
41 #
42 # Directory hierarchies to skip completely
43 SKIP ^usr/lib/libc/ # optimized libc
44 SKIP ^usr/lib/rcm/ # 4426119
45 SKIP ^usr/perl5/ # alan's taking care of these :-
46 SKIP ^usr/src/ # no objects in source code
47 #
48 # Individual files that we don't examine
49 SKIP ^boot/grub/bin/grub$
50 SKIP ^usr/apache/libexec/mod_ipp.so$ # Apache loadable module
51 # USIII specific extns. cause ldd noise on USII bld. m/c
52 SKIP ^usr/lib/fps/sun4u/UltraSPARC.*/*fptest$
53 SKIP ^usr/MACH(lib)/lddstub$ # lddstub has no dependencies
54 SKIP ^usr/MACH(lib)/libssagent\.\so\.\1$ # 4328854
55 SKIP ^usr/lib/MACH(iconv)/geniconvtbl.so$ # 4384329
56 #
57 # picl file exclusions (4385799)
58 SKIP ^usr/platform./libsvcpplugin_psr\.\so\.\1
59 SKIP ^usr/platform./libsvcpolicy_psr\.\so\.\1
60 SKIP ^usr/platform./libsvcpolicy\.\so\.\1

```

new/exception\_lists/check\_rtime

2

```

61 SKIP ^usr/lib/sysevent/modules/picl_slm.so$
62 #
63 # Objects that are allowed to have executable data segments
64 EXEC_DATA ^MACH(lib)/ld\.\so\.\1$
65 EXEC_DATA ^lib/libc\.\so\.\1$ # 6524709, 32-bit, needed for x86 only
66 EXEC_DATA ^lib/amd64/libumem\.\so\.\1$ # ptcumem
67 EXEC_DATA ^lib/libumem\.\so\.\1$ # ptcumem
68 EXEC_DATA ^opt/SUNWdtrt/tst./ustack/tst\.\helper\.\exe$
69 EXEC_DATA ^platform./MACH(kernel)/unix$
70 EXEC_DATA ^platform./multiboot$
71 EXEC_DATA ^MACH(lib)/libumem_trampoline\.\so\.\1$
72 #
73 # Objects that are allowed to have an executable stack
74 EXEC_STACK ^platform./MACH(kernel)/unix$
75 EXEC_STACK ^platform./multiboot$
76 #
77 # Objects for which we allow relocations to the text segment
78 TEXTREL ^platform./MACH(kernel)/unix$
79 #
80 # Directories and files that are allowed to have no direct bound symbols
81 NODIRECT ^platform./MACH(kernel)/unix$
82 NODIRECT ^usr/ucb
83 NODIRECT ^usr/4lib/sbcp$
84 #
85 # Identify any files that should be skipped when building a crle(1)
86 # configuration file. As the hwcab libraries can be loop-back mounted onto
87 # libc, these can confuse crle(1) because of their identical dev/inode.
88 NOCRLEALT ^usr/lib/libc/libc_hwcab[1-3].so.1$
89 #
90 # Files that should contain debugging information.
91 STAB ^platform./MACH(kernel)/unix$
92 #
93 # Files that are allowed undefined references
94 UNDEF_REF ^usr/lib/libnisdb\.\so\.\2$
95 UNDEF_REF ^usr/snadm/lib/libsvm\.\so\.\1$
96 #
97 # Objects allowed to have unused dependencies
98 UNUSED_DEPS ^usr/lib/picl/plugins/ # require devtree dependencies
99 UNUSED_DEPS ^usr/lib/libp # profile libc makes libm an unused dep of libc
100 #
101 # libm.so.2 dependency
102 UNUSED_OBJ unused object=.*MACH(lib)/libm_hwcab\.\so\.\2
103 #
104 # Unused runpaths due to dlopen() use
105 UNUSED_RPATH /usr/lib/fs/autofs.*\ from\ .automountd
106 UNUSED_RPATH /etc/ppp/plugins.*\ from\ .*pppd
107 UNUSED_RPATH /usr/lib/inet/ppp.*\ from\ .*pppd
108 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libipsecutil\.\so\.\1
109 UNUSED_RPATH /usr/platform./rsmlib.*\ from\ .*librsmlib\.\so\.\2
110 UNUSED_RPATH \${ORIGIN}.*\ from\ .*fcode.so
111 UNUSED_RPATH /opt/VRTSvxvm/lib.*\ from\ .*libdiskmgt\.\so\.\1
112 #
113 # Unused runpaths in picl code
114 UNUSED_RPATH /usr/platform./.*\ from\ ./usr/platform
115 UNUSED_RPATH /usr/lib/picl.*\ from\ ./usr/platform
116 UNUSED_RPATH /usr/platform./.*\ from\ ./usr/lib/picl
117 #
118 # Unused runpaths in non-OSNET objects we can't change
119 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libnss3\.\so
120 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libnssutil3\.\so
121 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libsmime3\.\so
122 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libssl3\.\so
123 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libdbus-1\.\so\.\3
124 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libdbus-glib-1\.\so\.\2
125 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libglib-2\.\0\.\so\.\0
126 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libglib-2\.\0\.\so\.\0

```

## new/exception\_lists/check\_rtime

3

```

127 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libgobject-2\0\0\so\0
128 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libgobject-2\0\0\so\0
129 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libgthread-2\0\0\so\0
130 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libgthread-2\0\0\so\0
131 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libcrypto\so\0\9\8
132 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libnetsmp\so\15
133 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libgcc_s\so\1
134 UNUSED_RPATH /usr/ccs/lib.*\ from\ .*libgcc_s\so\1
135 UNUSED_RPATH /usr/lib.*\ from\ .*libgcc_s\so\1
136 UNUSED_RPATH /usr/postgres/8.3/lib.*\ from\ .*libpq\so\5
137 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libpq\so\5
138 UNUSED_RPATH /usr/lib.*\ from\ ./usr/lib/mps
139 UNUSED_RPATH /usr/ccs/lib.*\ from\ ./usr/lib/mps
140 UNUSED_RPATH /usr/gnu/lib.*\ from\ .*usr/lib/libpython2\6
141 UNUSED_RPATH /usr/snadm/lib.*\ from\ .*usr/snadm/lib/libspmicommon\so\1
142 #endif /* ! codereview */

144 # Unused runpaths for reasons not captured above
145 UNUSED_RPATH /usr/lib/smbstr.*\ from\ .*libsmb\so\1 # future needs
146 UNUSED_RPATH /usr.*\ from\ .*tst\gcc\exe # gcc built

149 # Unreferenced objects of non-OSnet objects we can't change
150 UNREF_OBJ /lib.*\ of\ .*libcimapi\so
151 UNREF_OBJ /lib.*\ of\ .*libdbus-1\so\3
152 UNREF_OBJ /lib.*\ of\ .*libdbus-glib-1\so\2
153 UNREF_OBJ /lib.*\ of\ .*libgio-2.0\so\0
154 UNREF_OBJ /lib.*\ of\ .*libglib-2.0\so\0
155 UNREF_OBJ /lib.*\ of\ .*libgobject-2.0\so\0
156 UNREF_OBJ /lib.*\ of\ .*libgthread-2.0\so\0
157 UNREF_OBJ /lib.*\ of\ .*libjvm\so
158 UNREF_OBJ /lib.*\ of\ .*libnetsmp\so\30
159 UNREF_OBJ /lib.*\ of\ .*libnetsnmpagent\so\30
160 UNREF_OBJ /lib.*\ of\ .*libnetsnmpmibs\so\30
161 UNREF_OBJ /lib.*\ of\ .*libnetsnmphelpers\so\30
162 UNREF_OBJ /lib.*\ of\ .*libnspr4\so
163 UNREF_OBJ /lib.*\ of\ .*libpq\so\5
164 UNREF_OBJ /lib.*\ of\ .*libsoftokn3\so
165 UNREF_OBJ /lib.*\ of\ .*libspmicommon\so\1
166 UNREF_OBJ /lib.*\ of\ .*libspmocommon\so\1
167 UNREF_OBJ /lib.*\ of\ .*libssl3\so
168 UNREF_OBJ /lib.*\ of\ .*libtspi\so\1
169 UNREF_OBJ /lib.*\ of\ .*libxml2\so\2
170 UNREF_OBJ /lib.*\ of\ .*libxslt\so\1
171 UNREF_OBJ /lib.*\ of\ .*libpq\so\4
172 UNREF_OBJ /lib.*\ of\ .*libpython2\4\so\1\0
173 UNREF_OBJ /lib.*\ of\ .*libpython2\6\so\1\0
174 UNREF_OBJ /libgcc_s.*\ of\ .*libstdc\+\+\so\6

176 # Unreferenced object of objects we can't change for other reasons
177 UNREF_OBJ /libmapallocc\so\1;\ unused\ dependency\ of # interposer
178 UNREF_OBJ /libstdc\+\+\so\6;\ unused\ dependency\ of # gcc build
179 UNREF_OBJ /libm\so\2.*\ of\ .*libstdc\+\+\so\6 # gcc build
180 UNREF_OBJ /libgcc_s\so\1.*\ of\ .*libstdc\+\+\so\6 # omnios gcc mix
181 UNREF_OBJ /lib.*\ of\ .*lib/picl/plugins/ # picl
182 UNREF_OBJ /lib.*\ of\ .*kcfid # interposer
183 UNREF_OBJ /libpkcs11\so\1;.*\ of\ .*libkmf\so\1 # interposed
184 # Referenced by the Studio build, not the GCC build. GCC eliminates the unused
185 # statics which have the dependence.
186 UNREF_OBJ /lib\so\1.*\ of\ .*kldap\so\1

189 # Objects that used to contain system functionality that has since
190 # migrated to libc. We preserve these libraries as pure filters for
191 # backward compatability but nothing needs to link to them.
192 OLDDEP libaio\so\1 # onnv build 44

```

## new/exception\_lists/check\_rtime

4

```

193 OLDDEP libdl\so\1 # on10 build 49
194 OLDDEP libdoor\so\1 # onnv build 12
195 OLDDEP libintl\so\1 # on297 build 7
196 OLDDEP libpthread\so\1 # on10 build 53
197 OLDDEP librt\so\1 # onnv build 44
198 OLDDEP libsched\so\1 # on10 build 36
199 OLDDEP libthread\so\1 # on10 build 53
200 OLDDEP libw\so\1 # on297 build 7

202 # Files for which we skip checking of duplicate addresses in the
203 # symbol sort sections. Such exceptions should be rare --- most code will
204 # not have duplicate addresses, since it takes assembler or a "#pragma weak"
205 # to do such aliasing in C. C++ is different: The compiler generates aliases
206 # for implementation reasons, and the mangled names used to encode argument
207 # and return value types are difficult to handle well in mapfiles.
208 # Furthermore, the Sun compiler and gcc use different and incompatible
209 # name mangling conventions. Since ON must be buildable by either, we
210 # would have to maintain two sets of mapfiles for each such object.
211 # C++ use is rare in ON, so this is not worth pursuing.
212 #
213 NOSYMSORT opt/SUNWdtrt/tst/common/pid/tst.weak2.exe # DTrace test
214 NOSYMSORT lib/amd64/libnsl\so\1 # C++
215 NOSYMSORT lib/sparcv9/libnsl\so\1 # C++
216 NOSYMSORT lib/sparcv9/libfru\so\1 # C++
217 NOSYMSORT usr/lib/lms # C++
218 NOSYMSORT ld\so\1 # libc_pic.a use
219 NOSYMSORT lib/libsun_fc\so\1 # C++
220 NOSYMSORT lib/amd64/libsun_fc\so\1 # C++
221 NOSYMSORT lib/sparcv9/libsun_fc\so\1 # C++
222 NOSYMSORT usr/lib/amd64/libfru\so\1 # C++

225 # The libprtdiag_psr.so.1 objects built under usr/src/lib/libprtdiag_psr
226 # are a family, all built using the same makefile, targeted at different
227 # sparc hardware variants. There are a small number of cases where this
228 # one size fits all approach causes an object to be linked against an
229 # unneeded library.
230 UNREF_OBJ lib/(libdevinfo|libcfdgadm)\so\1;.*\ of\ .*SUNW,Netra-CP2300/1

```

```

*****
2046 Thu Feb 13 22:45:13 2014
new/usr/src/cmd/boot/bootadm/Makefile
don't pass in lint libraries which no longer exist
gag lint errors that stem from lint libraries that the omnios build system does
*****
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 # Copyright 2011 Nexenta Systems, Inc. All rights reserved.
26 #

28 PROG= bootadm

30 SBINLINKS= $(PROG)

32 OBJS= bootadm.o bootadm_upgrade.o bootadm_hyper.o
33 SRCS = $(OBJS:.o=.c)

35 include ../Makefile.com

38 .KEEP_STATE:

40 LDLIBS_i386= -lfdisk
41 LDLIBS += -lnvpair -lgen -ladm -lefi -lscf -lbe -lzfs $(LDLIBS_$(MACH))

43 # There's no lint library for zlib, so only include this when building
44 $(PROG) := LDLIBS += -lz
45 LDLIBS += -lnvpair -lgen -ladm -lefi -lscf -lz -lbe -lzfs $(LDLIBS_$(MACH))

46 # Writing into string literals is incorrect. We need to match gcc's
47 # behavior, which causes us to take SIGSEGV on such a write.
48 CFLAGS += $(XSTRCONST)
49 CPPFLAGS += -D_FILE_OFFSET_BITS=64
50 CPPFLAGS += -I../..../uts/common -I../..../common

52 CERRWARN += -_gcc=-Wno-parentheses
53 CERRWARN += -_gcc=-Wno-uninitialized
54 CERRWARN += -_gcc=-Wno-unused-label

56 LINTFLAGS += -erroff=E_INCONS_ARG_DECL2
57 LINTFLAGS += -erroff=E_INCONS_VAL_TYPE_DECL2
58 LINTFLAGS += -erroff=E_NAME_DEF_NOT_USED2
59 LINTFLAGS += -erroff=E_NAME_USED_NOT_DEF2

```

```

60 #endif /* ! codereview */

62 all: $(PROG)

64 $(PROG): $(OBJS) bootadm.h
65         $(LINK.c) -o $@ $(OBJS) $(LDLIBS)
66         $(POST_PROCESS)

68 install: all $(ROOTSBINPROG) .WAIT $(ROOTUSRSBINLINKS)

70 clean:
71         -$(RM) $(OBJS)

73 _msg:

75 lint: lint_SRCS

77 include ../Makefile.targ

```

new/usr/src/cmd/boot/fiocompress/Makefile

1

```
*****
1527 Thu Feb 13 22:45:13 2014
new/usr/src/cmd/boot/fiocompress/Makefile
gag lint errors that stem from lint libraries that the omnios build system does
*****
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 #
25 # ident "%Z%M% %I% %E% SMI"
26 #

26 PROG= fiocompress

28 SBINLINKS= $(PROG)

30 OBJS= fiocompress.o
31 SRCS = $(OBJS:.o=.c)

33 include ../Makefile.com

35 .KEEP_STATE:

37 LDLIBS += -lz

39 CFLAGS += -I../../uts/common
40 LINTFLAGS += -I../../uts/common

42 # definitions for lint
43 # until libz is compiled against ON header files (uid/gid)
44 LINTFLAGS += -erroff=E_INCONS_ARG_DECL2
45 LINTFLAGS += -erroff=E_INCONS_VAL_TYPE_DECL2
46 LINTFLAGS += -erroff=E_NAME_USED_NOT_DEF2
47 #endif /* ! codereview */

49 all: $(PROG)

51 $(PROG): $(OBJS)
52 $(LINK.c) -o $@ $(OBJS) $(LDLIBS)
53 $(POST_PROCESS)

55 install: all $(ROOTSBINPROG) .WAIT $(ROOTUSRSBINLINKS)

57 clean:
58 -$(RM) $(OBJS)
```

new/usr/src/cmd/boot/fiocompress/Makefile

2

```
60 _msg:
62 lint: lint_SRCS
64 include ../Makefile.targ
```

new/usr/src/cmd/cmd-crypto/tpmadm/Makefile

1

```
*****
1566 Thu Feb 13 22:45:14 2014
new/usr/src/cmd/cmd-crypto/tpmadm/Makefile
don't pass in lint libraries which no longer exist
*****
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 2009 Sun Microsystems, Inc. All rights reserved.
22 # Use is subject to license terms.
23 #

25 PROG = tpmadm

27 OBJS = main.o \
28       admin_cmds.o

30 include ../../Makefile.cmd

32 SRCS = $(OBJS:%.o=%.c)
33 POFILES = $(OBJS:%.o=%.po)
34 POFILE = $(PROG)_msg.po
35 MSGFILES=$(SRCS:%.c=%.i)

37 CPPFLAGS += -xCC
38 LINTFLAGS += -XCC -L$(ADJUNCT_PROTO)/usr/lib
39 LDLIBS += -luuid

41 # libtspi has no lint library, so we can only include this while building
42 $(PROG) := LDLIBS += -ltspi
43 LDLIBS += -ltspi -luuid

44 CLOBBERFILES = C.ln $(POFILE) $(POFILES)

46 .KEEP_STATE:

48 all : $(PROG)

50 $(PROG) : $(OBJS)
51         $(LINK.c) -o $@ $(OBJS) $(DYNFLAGS) $(LDLIBS)
52         $(POST_PROCESS)

54 $(POFILE) : $(POFILES)
55         $(RM) $@; $(CAT) $(POFILES) > $@

57 install : all $(ROOTPROG)

59 clean :
60         $(RM) $(OBJS)
```

new/usr/src/cmd/cmd-crypto/tpmadm/Makefile

2

```
62 lint : lint_SRCS
64 include ../../Makefile.targ
```

new/usr/src/cmd/cmd-inet/usr.lib/wanboot/pl2split/Makefile

1

```
*****
1265 Thu Feb 13 22:45:14 2014
new/usr/src/cmd/cmd-inet/usr.lib/wanboot/pl2split/Makefile
don't pass in lint libraries which no longer exist
gag lint errors that stem from lint libraries that the omnios build system does
*****
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 2009 Sun Microsystems, Inc. All rights reserved.
22 # Use is subject to license terms.
23 #

25 include ../Makefile.com

27 PROG=          pl2split
28 LDLIBS +=      -lwanboot -linetutil -lwanbootutil
28 LDDLIBS +=     -lwanboot -linetutil -lwanbootutil -lcrypto
29 CPPFLAGS +=    -I$(CMNCRYPTDIR)

31 # libcrypto has no lint library, so we can only include this while building
32 $(PROG) := LDLIBS += -lcrypto

34 LINTFLAGS +=   -erroff=E_NAME_USED_NOT_DEF2

36 #endif /* ! codereview */
37 all:           $(PROG)

39 install:      all $(ROOTCMD)

41 clean:

43 lint:         lint_PROG

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

```

*****
15474 Thu Feb 13 22:45:15 2014
new/usr/src/cmd/cmd-inet/usr.lib/wanboot/pl2split/pl2split.c
remove lint supression made unnecessary by the openssl upgrade
*****
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"

27 #include <stdio.h>
28 #include <libintl.h>
29 #include <locale.h>
30 #include <sys/types.h>
31 #include <sys/stat.h>
32 #include <sys/wanboot_impl.h>
33 #include <unistd.h>
34 #include <string.h>
35 #include <libinetutil.h>
36 #include <wanbootutil.h>

38 #include <openssl/crypto.h>
39 #include <openssl/buffer.h>
40 #include <openssl/bio.h>
41 #include <openssl/err.h>
42 #include <openssl/x509.h>
43 #include <openssl/x509v3.h>
44 #include <openssl/pkcs12.h>
45 #include <openssl/evp.h>
46 #include <pl2aux.h>

48 static boolean_t verbose = B_FALSE;      /* When nonzero, do in verbose mode */

50 /* The following match/cert values require PKCS12 */
51 static int  matchty;      /* Type of matching do to on input */
52 static char *k_matchval;  /* localkeyid value to match */
53 static uint_t k_len;      /* length of k_matchval */

55 #define IO_KEYFILE      1      /* Have a separate key file or data */
56 #define IO_CERTFILE    2      /* Have a separate cert file or data */
57 #define IO_TRUSTFILE   4      /* Have a separate trustanchor file */

59 static char *input = NULL;      /* Consolidated input file */

```

```

60 static char *key_out = NULL;      /* Key file to be output */
61 static char *cert_out = NULL;     /* Cert file to be output */
62 static char *trust_out = NULL;    /* Trust anchor file to be output */
63 static uint_t outfiles;           /* What files are there for output */
64 static char *progname;

66 /* Returns from time_check */
67 typedef enum {
68     CHK_TIME_OK = 0,                /* Cert in effect and not expired */
69     CHK_TIME_BEFORE_BAD,           /* not_before field is invalid */
70     CHK_TIME_AFTER_BAD,           /* not_after field is invalid */
71     CHK_TIME_IS_BEFORE,           /* Cert not yet in force */
72     CHK_TIME_HAS_EXPIRED          /* Cert has expired */
73 } time_errs_t;
_____ unchanged_portion_omitted _____

219 static int
220 do_certs(void)
221 {
222     char *bufp;
223     STACK_OF(X509) *ta_in = NULL;
224     EVP_PKEY *pkey_in = NULL;
225     X509 *xcert_in = NULL;

227     sunw_crypto_init();

229     if (read_files(&ta_in, &xcert_in, &pkey_in) < 0)
230         return (-1);

232     if (verbose) {
233         if (xcert_in != NULL) {
234             (void) printf(gettext("\nMain cert:\n"));

236             /*
237              * sunw_subject_attrs() returns a pointer to
238              * memory allocated on our behalf. The same
239              * behavior is exhibited by sunw_issuer_attrs().
240              */
241             bufp = sunw_subject_attrs(xcert_in, NULL, 0);
242             if (bufp != NULL) {
243                 (void) printf(gettext(" Subject: %s\n"),
244                     bufp);
245                 OPENSSL_free(bufp);
246             }

248             bufp = sunw_issuer_attrs(xcert_in, NULL, 0);
249             if (bufp != NULL) {
250                 (void) printf(gettext(" Issuer: %s\n"), bufp);
251                 OPENSSL_free(bufp);
252             }

254             (void) sunw_print_times(stdout, PRNT_BOTH, NULL,
255                 xcert_in);
256         }

258         if (ta_in != NULL) {
259             X509 *x;
260             int i;

262             for (i = 0; i < sk_X509_num(ta_in); i++) {
263                 /* LINTED */
264                 x = sk_X509_value(ta_in, i);
265                 (void) printf(
266                     gettext("\nTrust Anchor cert %d:\n"), i);

267             /*

```

```

268     * sunw_subject_attrs() returns a pointer to
269     * memory allocated on our behalf. We get the
270     * same behavior from sunw_issuer_attrs().
271     */
272     bufp = sunw_subject_attrs(x, NULL, 0);
273     if (bufp != NULL) {
274         (void) printf(
275             gettext(" Subject: %s\n"), bufp);
276         OPENSSL_free(bufp);
277     }

279     bufp = sunw_issuer_attrs(x, NULL, 0);
280     if (bufp != NULL) {
281         (void) printf(
282             gettext(" Issuer: %s\n"), bufp);
283         OPENSSL_free(bufp);
284     }

286     (void) sunw_print_times(stdout, PRNT_BOTH,
287         NULL, x);
288     }
289 }
290

292 check_certs(ta_in, &xcert_in);
293 if (xcert_in != NULL && pkey_in != NULL) {
294     if (sunw_check_keys(xcert_in, pkey_in) == 0) {
295         wbku_printrerr("warning: key and certificate do "
296             "not match\n");
297     }
298 }

300 return (write_files(ta_in, xcert_in, pkey_in));
301 }

```

unchanged portion omitted

```

354 static void
355 check_certs(STACK_OF(X509) *ta_in, X509 **c_in)
356 {
357     X509 *curr;
358     time_errs_t ret;
359     int i;
360     int del_expired = (outfiles != 0);

362     if (c_in != NULL && *c_in != NULL) {
363         ret = time_check_print(*c_in);
364         if ((ret != CHK_TIME_OK && ret != CHK_TIME_IS_BEFORE) &&
365             del_expired) {
366             (void) fprintf(stderr, gettext(" Removing cert\n"));
367             X509_free(*c_in);
368             *c_in = NULL;
369         }
370     }

372     if (ta_in == NULL)
373         return;

375     for (i = 0; i < sk_X509_num(ta_in); ) {
376         /* LINTED */
377         curr = sk_X509_value(ta_in, i);
378         ret = time_check_print(curr);
379         if ((ret != CHK_TIME_OK && ret != CHK_TIME_IS_BEFORE) &&
380             del_expired) {
381             (void) fprintf(stderr, gettext(" Removing cert\n"));
382             /* LINTED */
383             curr = sk_X509_delete(ta_in, i);

```

```

382         X509_free(curr);
383         continue;
384     }
385     i++;
386 }
387 }

```

unchanged portion omitted

```

521 static int
522 do_ofile(char *name, EVP_PKEY *pkey, X509 *cert, STACK_OF(X509) *ta)
523 {
524     STACK_OF(EVP_PKEY) *klist = NULL;
525     STACK_OF(X509) *clist = NULL;
526     PKCS12 *p12 = NULL;
527     int ret = 0;
528     FILE *fp;
529     struct stat sbuf;

531     if (stat(name, &sbuf) == 0 && !S_ISREG(sbuf.st_mode)) {
532         wbku_printrerr("%s is not a regular file\n", name);
533         return (-1);
534     }

536     if ((fp = fopen(name, "w")) == NULL) {
537         wbku_printrerr("cannot open output file %s", name);
538         return (-1);
539     }

541     if ((clist = sk_X509_new_null()) == NULL ||
542         (klist = sk_EVP_PKEY_new_null()) == NULL) {
543         wbku_printrerr("out of memory\n");
544         ret = -1;
545         goto cleanup;
546     }

548     if (cert != NULL && sk_X509_push(clist, cert) == 0) {
549         wbku_printrerr("out of memory\n");
550         ret = -1;
551         goto cleanup;
552     }

554     if (pkey != NULL && sk_EVP_PKEY_push(klist, pkey) == 0) {
555         wbku_printrerr("out of memory\n");
556         ret = -1;
557         goto cleanup;
558     }

560     p12 = sunw_PKCS12_create(WANBOOT_PASSPHRASE, klist, clist, ta);
561     if (p12 == NULL) {
562         wbku_printrerr("cannot create %s: %s\n", name, cryptoerr());
563         ret = -1;
564         goto cleanup;
565     }

567     if (i2d_PKCS12_fp(fp, p12) == 0) {
568         wbku_printrerr("cannot write %s: %s\n", name, cryptoerr());
569         ret = -1;
570         goto cleanup;
571     }

573 cleanup:
574     (void) fclose(fp);
575     if (p12 != NULL)
576         PKCS12_free(p12);
577     /*
578      * Put the cert and pkey off of the stack so that they won't

```



```
579     * be freed two times. (If they get left in the stack then
580     * they will be freed with the stack.)
581     */
582     if (clist != NULL) {
583         if (cert != NULL && sk_X509_num(clist) == 1) {
584             /* LINTED */
585             (void) sk_X509_delete(clist, 0);
586         }
587         sk_X509_pop_free(clist, X509_free);
588     }
589     if (klist != NULL) {
590         if (pkey != NULL && sk_EVP_PKEY_num(klist) == 1) {
591             /* LINTED */
592             (void) sk_EVP_PKEY_delete(klist, 0);
593         }
594         sk_EVP_PKEY_pop_free(klist, sunw_evpc_pkey_free);
595     }
596     return (ret);
597 }
598 _____unchanged_portion_omitted_____
```

new/usr/src/cmd/cmd-inet/usr.lib/wanboot/wanboot-cgi/Makefile

1

\*\*\*\*\*

1236 Thu Feb 13 22:45:15 2014

new/usr/src/cmd/cmd-inet/usr.lib/wanboot/wanboot-cgi/Makefile

don't pass in lint libraries which no longer exist

\*\*\*\*\*

```
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 2009 Sun Microsystems, Inc. All rights reserved.
22 # Use is subject to license terms.
23 #
```

```
25 include ../Makefile.com
```

```
27 PROG = wanboot-cgi
28 LDLIBS += -lgen -lnsl -lwanbootutil -lnvpair -lwanboot
28 LDLIBS += -lgen -lnsl -lwanbootutil -lnvpair -lwanboot -lcrypto
29 CPPFLAGS += -I$(CMNCRYPTDIR)
```

```
31 # libcrypto has no lint library, so we can only include this while building
```

```
32 $(PROG) := LDLIBS += -lcrypto
```

```
34 #endif /* ! codereview */
```

```
35 all: $(PROG)
```

```
37 install: all $(ROOTCMD)
```

```
39 clean:
```

```
41 lint: lint_PROG
```

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

```

*****
45017 Thu Feb 13 22:45:16 2014
new/usr/src/cmd/cmd-inet/usr.lib/wanboot/wanboot-cgi/wanboot-cgi.c
remove lint supression made unnecessary by the openssl upgrade
*****
_____unchanged_portion_omitted_____

773 /*
774  * Add the certs found in the trustfile found in path (a trust store) to
775  * the file found at bootfs_dir/truststore.  If necessary, create the
776  * output file.
777  */
778 static int
779 build_trustfile(const char *path, void *truststorepath)
780 {
781     int             ret = WBCGI_FTW_CBERR;
782     STACK_OF(X509) *i_anchors = NULL;
783     STACK_OF(X509) *o_anchors = NULL;
784     char            message[WBCGI_MAXBUF];
785     PKCS12          *p12 = NULL;
786     FILE            *rfp = NULL;
787     FILE            *wfp = NULL;
788     struct stat     i_st;
789     struct stat     o_st;
790     X509            *x = NULL;
791     int             errtype = 0;
792     int             wfd = -1;
793     int             chars;
794     int             i;

796     if (!WBCGI_FILE_EXISTS(path, i_st)) {
797         goto cleanup;
798     }

800     if (WBCGI_FILE_EXISTS((char *)truststorepath, o_st)) {
801         /*
802          * If we are inadvertently writing to the input file.
803          * return success.
804          * XXX Pete: how can this happen, and why success?
805          */
806         if (i_st.st_ino == o_st.st_ino) {
807             ret = WBCGI_FTW_CBCONT;
808             goto cleanup;
809         }
810         if ((wfp = fopen((char *)truststorepath, "r+")) == NULL) {
811             goto cleanup;
812         }
813         /*
814          * Read what's already there, so that new information
815          * can be added.
816          */
817         if ((p12 = d2i_PKCS12_fp(wfp, NULL)) == NULL) {
818             errtype = 1;
819             goto cleanup;
820         }
821         i = sunw_PKCS12_parse(p12, WANBOOT_PASSPHRASE, DO_NONE, NULL,
822             0, NULL, NULL, NULL, &o_anchors);
823         if (i <= 0) {
824             errtype = 1;
825             goto cleanup;
826         }

828         PKCS12_free(p12);
829         p12 = NULL;
830     } else {
831         if (errno != ENOENT) {

```

```

832         chars = sprintf(message, sizeof (message),
833             "(error accessing file %s, error %s)",
834             path, strerror(errno));
835         if (chars > 0 && chars < sizeof (message))
836             print_status(500, message);
837     else
838         print_status(500, NULL);
839     return (WBCGI_FTW_CBERR);
840 }

842 /*
843  * Note: We could copy the file to the new trustfile, but
844  * we can't verify the password that way.  Therefore, copy
845  * it by reading it.
846  */
847     if ((wfd = open((char *)truststorepath,
848         O_CREAT|O_EXCL|O_RDWR, 0700)) < 0) {
849         goto cleanup;
850     }
851     if ((wfp = fdopen(wfd, "w+")) == NULL) {
852         goto cleanup;
853     }
854     o_anchors = sk_X509_new_null();
855     if (o_anchors == NULL) {
856         goto cleanup;
857     }
858 }

860     if ((rfp = fopen(path, "r")) == NULL) {
861         goto cleanup;
862     }
863     if ((p12 = d2i_PKCS12_fp(rfp, NULL)) == NULL) {
864         errtype = 1;
865         goto cleanup;
866     }
867     i = sunw_PKCS12_parse(p12, WANBOOT_PASSPHRASE, DO_NONE, NULL, 0, NULL,
868         NULL, NULL, &i_anchors);
869     if (i <= 0) {
870         errtype = 1;
871         goto cleanup;
872     }
873     PKCS12_free(p12);
874     p12 = NULL;

876     /*
877      * Merge the two stacks of pkcs12 certs.
878      */
879     for (i = 0; i < sk_X509_num(i_anchors); i++) {
880         /* LINTED */
881         x = sk_X509_delete(i_anchors, i);
882         (void) sk_X509_push(o_anchors, x);
883     }

884     /*
885      * Create the pkcs12 structure from the modified input stack and
886      * then write out that structure.
887      */
888     p12 = sunw_PKCS12_create((const char *)WANBOOT_PASSPHRASE, NULL, NULL,
889         o_anchors);
890     if (p12 == NULL) {
891         goto cleanup;
892     }
893     rewind(wfp);
894     if (i2d_PKCS12_fp(wfp, p12) == 0) {
895         goto cleanup;
896     }

```

```

898     ret = WBCGI_FTW_CBCONT;
899 cleanup:
900     if (ret == WBCGI_FTW_CBERR) {
901         if (errtype == 1) {
902             chars = snprintf(message, sizeof (message),
903                 "(internal PKCS12 error while copying %s to %s)",
904                 path, (char *)truststorepath);
905         } else {
906             chars = snprintf(message, sizeof (message),
907                 "(error copying %s to %s)",
908                 path, (char *)truststorepath);
909         }
910         if (chars > 0 && chars <= sizeof (message)) {
911             print_status(500, message);
912         } else {
913             print_status(500, NULL);
914         }
915     }
916     if (rfp != NULL) {
917         (void) fclose(rfp);
918     }
919     if (wfp != NULL) {
920         /* Will also close wfd */
921         (void) fclose(wfp);
922     }
923     if (p12 != NULL) {
924         PKCS12_free(p12);
925     }
926     if (i_anchors != NULL) {
927         sk_X509_pop_free(i_anchors, X509_free);
928     }
929     if (o_anchors != NULL) {
930         sk_X509_pop_free(o_anchors, X509_free);
931     }
932
933     return (ret);
934 }
_____ unchanged_portion_omitted _____

1056 /*
1057  * Loop through the certificates in a file
1058  */
1059 static int
1060 get_hostnames(const char *path, void *nvl)
1061 {
1062     int             ret = WBCGI_FTW_CBERR;
1063     STACK_OF(X509) *certs = NULL;
1064     PKCS12          *p12 = NULL;
1065     char            message[WBCGI_MAXBUF];
1066     char            buf[WBCGI_MAXBUF + 1];
1067     FILE            *rfp = NULL;
1068     X509            *x = NULL;
1069     int             errtype = 0;
1070     int             chars;
1071     int             i;

1073     if ((rfp = fopen(path, "r")) == NULL) {
1074         goto cleanup;
1075     }

1077     if ((p12 = d2i_PKCS12_fp(rfp, NULL)) == NULL) {
1078         errtype = 1;
1079         goto cleanup;
1080     }
1081     i = sunw_PKCS12_parse(p12, WANBOOT_PASSPHRASE, DO_NONE, NULL, 0, NULL,

```

```

1082         NULL, NULL, &certs);
1083     if (i <= 0) {
1084         errtype = 1;
1085         goto cleanup;
1086     }

1088     PKCS12_free(p12);
1089     p12 = NULL;

1091     for (i = 0; i < sk_X509_num(cert); i++) {
1092         /* LINTED */
1093         x = sk_X509_value(cert, i);
1094         if (!one_name(sunw_issuer_attrs(x, buf, sizeof (buf) - 1),
1095             nvl)) {
1096             goto cleanup;
1097         }
1098     }

1099     ret = WBCGI_FTW_CBCONT;
1100 cleanup:
1101     if (ret == WBCGI_FTW_CBERR) {
1102         if (errtype == 1) {
1103             chars = snprintf(message, sizeof (message),
1104                 "(internal PKCS12 error reading %s)", path);
1105         } else {
1106             chars = snprintf(message, sizeof (message),
1107                 "error reading %s", path);
1108         }
1109         if (chars > 0 && chars <= sizeof (message)) {
1110             print_status(500, message);
1111         } else {
1112             print_status(500, NULL);
1113         }
1114     }
1115     if (rfp != NULL) {
1116         (void) fclose(rfp);
1117     }
1118     if (p12 != NULL) {
1119         PKCS12_free(p12);
1120     }
1121     if (certs != NULL) {
1122         sk_X509_pop_free(cert, X509_free);
1123     }

1125     return (ret);
1126 }
_____ unchanged_portion_omitted _____

```

new/usr/src/cmd/fm/modules/common/fabric-xlate/Makefile

1

```
*****
1517 Thu Feb 13 22:45:16 2014
new/usr/src/cmd/fm/modules/common/fabric-xlate/Makefile
don't pass in lint libraries which no longer exist
*****
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 MODULE = fabric-xlate
27 CLASS = common
28 SRCS = fabric-xlate.c fx_epkt.c fx_fabric.c fx_fire.c fx_subr.c

30 include ../../Makefile.plugin

32 CPPFLAGS += -I$(ADJUNCT_PROTO)/usr/include/libxml2 -I$(KMFDIR)/include -I.
33 CPPFLAGS += -I$(SRC)/uts/sun4v/io/px
34 INCDIRS = $(SRC)/uts/common
35 CFLAGS += -I$(INCDIRS)
36 LINTFLAGS += -I$(INCDIRS)

38 LDFLAGS += -L$(ROOT)/usr/lib/fm -R/usr/lib/fm
39 LDLIBS += -luutil -L$(ROOTLIB)/fm -ltopo

41 # libxml2 has no lint library, so we can only include this while building
42 $(PROG) := LDLIBS += -lxml2
39 LDLIBS += -luutil -lxml2 -L$(ROOTLIB)/fm -ltopo

44 CERRWARN += -_gcc=-Wno-parentheses
45 CERRWARN += -_gcc=-Wno-switch
46 CERRWARN += -_gcc=-Wno-uninitialized
```

```

*****
1699 Thu Feb 13 22:45:17 2014
new/usr/src/cmd/isns/isnsadm/Makefile
don't pass in lint libraries which no longer exist
gag lint errors that stem from lint libraries that the omnios build system does
*****
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

21 #
22 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #

26 PROG = isnsadm
27 SRCS = cmdparse.c isnsadm.c isnsadm_msg.c
28 OBJS = $(SRCS:%.c=%.o)

30 include ../../Makefile.cmd

32 LDLIBS += -lscf -lc
32 LDLIBS += -lxml2 -lscf -lc
33 CFLAGS += $(CCVERBOSE)
34 CPPFLAGS += -I. -I$(ADJUNCT_PROTO)/usr/include/libxml2 -I../isnsd
35 POFILE = isnsadm_all.po
36 POFILES = $(OBJS:%.o=%.po)

38 # libxml2 has no lint library, so we can only include this while building
39 $(PROG) := LDLIBS += -lxml2

41 #endif /* ! codereview */
42 LINTFLAGS += -xeroff=E_BAD_FORMAT_STR2
43 LINTFLAGS += -eroff=E_NAME_USED_NOT_DEF2
44 #endif /* ! codereview */

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

49 .KEEP_STATE:

51 all: $(PROG)

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

56 lint: lint_SRCS

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

```

```

60     $(POST_PROCESS)

62 $(POFILE): $(POFILES)
63     $(RM) $@
64     $(CAT) $(POFILES) > $@

66 include ../../Makefile.targ

68 install: all .WAIT $(ROOTUSRSBINPROG)

```

new/usr/src/cmd/isns/isnsd/Makefile

1

```
*****
2761 Thu Feb 13 22:45:17 2014
new/usr/src/cmd/isns/isnsd/Makefile
don't pass in lint libraries which no longer exist
*****
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 #
26 #
27 # Copyright (c) 2012 by Delphix. All rights reserved.
28 #
29 #
30 PROG = isns
31 DSRC = isns_provider.d
32 DTRACE_HEADER = $(DSRC:%.d=%.h)
33
34
35 COBJS = main.o qry.o admintf.o dsapi.o log.o sched.o dseng.o scn.o cache.o
36 COBJS += dump.o msgq.o server.o config.o esi.o obj.o utils.o dd.o func.o
37 COBJS += pdu.o door.o htable.o
38 OBJS = $(COBJS) $(DSRC:%.d=%.o)
39 SRCS = $(COBJS:%.o=%.c)
40 POFILE= isns.po
41 POFILES = $(COBJS:%.o=%.po)
42
43 include ../../Makefile.cmd
44
45 DATA_STORE = xml
46 XML_DIR = xml_def
47 ISNS_SVC = $(XML_DIR)/isns_server.xml
48
49 MANIFEST = isns_server.xml
50 ROOTMANIFESTDIR = $(ROOTSVCNETWORK)
51 $(ROOTSVCNETWORK)/isns_server.xml := FILEMODE = 0444
52
53 CFLAGS += $(CCVERBOSE)
54 CPPFLAGS += -I./ -I$(ADJUNCT_PROTO)/usr/include/libxml2
55 CPPFLAGS += -DTARGET_DATA_STORE=$(DATA_STORE)
56 CPPFLAGS += -D_POSIX_PTHREAD_SEMANTICS
57 LINTFLAGS += -xerroff=E_BAD_PTR_CAST_ALIGN
58 LINTFLAGS += -xerroff=E_NAME_USED_NOT_DEF2
59 LINTFLAGS += -xerroff=E_CASE_FALLTHRU
60
61 CERRWARN += _gcc=-Wno-char-subscripts
```

new/usr/src/cmd/isns/isnsd/Makefile

2

```
62 CERRWARN += _gcc=-Wno-ignored-qualifiers
63 CERRWARN += _gcc=-Wno-uninitialized
64 CERRWARN += _gcc=-Wno-implicit-function-declaration
65
66 CLOBBERFILES += $(DTRACE_HEADER)
67
68 obj.o := CERRWARN += -erroff=E_CONST_OBJ_SHOULD_HAVE_INITIZR
69 obj.o := CERRWARN += -erroff=E_IMPLICIT_DECL_FUNC_RETURN_INT
70
71 LDLIBS += -lsecdb -lsocket -lnsl -lscf
72
73 # libxml2 has no lint library, so we can only include this while building
74 $(PROG) := LDLIBS += -lxml2
75 LDLIBS += -lsecdb -lsocket -lnsl -lscf -lxml2
76
77 .KEEP_STATE:
78 all: $(PROG)
79
80 $(PROG): $(OBJS)
81 $(LINK.c) -o $@ $(OBJS) $(LDLIBS)
82 $(POST_PROCESS)
83
84 %.h: %.d
85 $(DTRACE) -h -s $< -o $@
86
87 %.o: %.c $(DTRACE_HEADER)
88 $(COMPILE.c) $<
89 $(POST_PROCESS_O)
90
91 %.o: %.d $(COBJS)
92 $(COMPILE.d) -s $< $(COBJS)
93 $(POST_PROCESS_O)
94
95 clean:
96 $(RM) $(OBJS)
97
98 lint: lint_SRCS
99
100 $(POFILE): $(POFILES)
101 $(RM) $@
102 $(CAT) $(POFILES) > $@
103
104 check: $(CHKMANIFEST)
105
106 include ../../Makefile.targ
107
108 install: all .WAIT $(ROOTUSRBINPROG) $(ROOTMANIFEST)
```

```

*****
22988 Thu Feb 13 22:45:18 2014
new/usr/src/cmd/mdb/i86pc/modules/unix/unix.c
LINTED a bad cast align warning that seems to be oddly omnios specific.
I think the alignment of any pointer we receive from mdb_zalloc must be
sufficiently aligned for 'long' on both x86 platforms.
*****
_____unchanged_portion_omitted_____

823 /*
824 * The x86 feature set is implemented as a bitmap array. That bitmap array is
825 * stored across a number of uchars based on the BT_SIZEOFMAP(NUM_X86_FEATURES)
826 * macro. We have the names for each of these features in unix's text segment
827 * so we do not have to duplicate them and instead just look them up.
828 */
829 /*ARGSUSED*/
830 static int
831 x86_featureset_cmd(uintptr_t addr, uint_t flags, int argc,
832                  const mdb_arg_t *argv)
833 {
834     uchar_t *fset;
835     GElf_Sym sym;
836     uintptr_t nptr;
837     char name[128];
838     int ii;

840     size_t sz = sizeof (uchar_t) * BT_SIZEOFMAP(NUM_X86_FEATURES);

842     if (argc != 0)
843         return (DCMD_USAGE);

845     if (mdb_lookup_by_name("x86_feature_names", &sym) == -1) {
846         mdb_warn("couldn't find x86_feature_names");
847         return (DCMD_ERR);
848     }

850     fset = mdb_zalloc(sz, UM_NOSLEEP);
851     if (fset == NULL) {
852         mdb_warn("failed to allocate memory for x86_featureset");
853         return (DCMD_ERR);
854     }

856     if (mdb_readvar(fset, "x86_featureset") != sz) {
857         mdb_warn("failed to read x86_featureset");
858         mdb_free(fset, sz);
859         return (DCMD_ERR);
860     }

862     for (ii = 0; ii < NUM_X86_FEATURES; ii++) {
863         /* LINTED */
864 #endif /* ! codereview */
865         if (!BT_TEST((ulong_t *)fset, ii))
866             continue;

868         if (mdb_vread(&nptr, sizeof (char *), sym.st_value +
869                    sizeof (void *) * ii) != sizeof (char *)) {
870             mdb_warn("failed to read feature array %d", ii);
871             mdb_free(fset, sz);
872             return (DCMD_ERR);
873         }

875         if (mdb_readstr(name, sizeof (name), nptr) == -1) {
876             mdb_warn("failed to read feature %d", ii);
877             mdb_free(fset, sz);
878             return (DCMD_ERR);
879         }

```

```

880         mdb_printf("%s\n", name);
881     }

883     mdb_free(fset, sz);
884     return (DCMD_OK);
885 }

887 static const mdb_dcmd_t dcmds[] = {
888     { "gate_desc", ":", "dump a gate descriptor", gate_desc },
889     { "idt", ":[-v]", "dump an IDT", idt },
890     { "ttrace", "[:-x]", "dump trap trace buffers", ttrace },
891     { "vatopfn", ":[-a as]", "translate address to physical page",
892       va2pfn_dcmd },
893     { "report_maps", ":[-m]",
894       "Given PFN, report mappings / page table usage",
895       report_maps_dcmd, report_maps_help },
896     { "htables", "", "Given hat_t *, lists all its htable_t * values",
897       htables_dcmd, htables_help },
898     { "ptable", ":[-m]", "Given PFN, dump contents of a page table",
899       ptable_dcmd, ptable_help },
900     { "pte", ":[-p XXXXX] [-l N]", "print human readable page table entry",
901       pte_dcmd },
902     { "pfntomfn", ":", "convert physical page to hypervisor machine page",
903       pfntomfn_dcmd },
904     { "mfntopfn", ":", "convert hypervisor machine page to physical page",
905       mfntopfn_dcmd },
906     { "memseg_list", ":", "show memseg list", memseg_list },
907     { "scalehrtm", ":",
908       "scale an unscaled high-res time", scalehrtm_cmd },
909     { "x86_featureset", NULL, "dump the x86_featureset vector",
910       x86_featureset_cmd },
911     { NULL }
912 };

914 static const mdb_walker_t walkers[] = {
915     { "ttrace", "walks trap trace buffers in reverse chronological order",
916       ttrace_walk_init, ttrace_walk_step, ttrace_walk_fini },
917     { "mutex_owner", "walks the owner of a mutex",
918       mutex_owner_init, mutex_owner_step },
919     { "memseg", "walk the memseg structures",
920       memseg_walk_init, memseg_walk_step, memseg_walk_fini },
921     { NULL }
922 };

924 static const mdb_modinfo_t modinfo = { MDB_API_VERSION, dcmds, walkers };

926 const mdb_modinfo_t *
927 _mdb_init(void)
928 {
929     return (&modinfo);
930 }

932 void
933 _mdb_fini(void)
934 {
935     free_mmu();
936 }

```



```

*****
14127 Thu Feb 13 22:45:18 2014
new/usr/src/cmd/smbdrv/smbd/smbd_spool.c
Correctly merge "2991 Allow building without SMB printing support"
*****
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) 2010, Oracle and/or its affiliates. All rights reserved.
23 * Copyright 2012 Nexenta Systems, Inc. All rights reserved.
24 */

26 /*
27  * CUPS support for the SMB and SPOOLSS print services.
28 */

30 #ifndef HAVE_CUPS
31 void smbd_load_printers(void) { return; }
32 void smbd_cups_init(void) { return; }
33 void smbd_cups_fini(void) { return; }
34 void smbd_spool_start(void) { return; }
35 void smbd_spool_stop(void) { return; }
36 #else

30 #include <sys/types.h>
31 #include <sys/stat.h>
32 #include <strings.h>
33 #include <syslog.h>
34 #include <signal.h>
35 #include <pthread.h>
36 #include <synch.h>
37 #include <dlfcn.h>
38 #include <errno.h>
39 #include <smbdrv/smb.h>
40 #include <smbdrv/smb_share.h>
41 #include "smbd.h"

43 #ifdef HAVE_CUPS
44 #endif /* ! codereview */
45 #include <cups/cups.h>

47 #define SMB_SPOOL_WAIT          2
48 #define SMBD_PJOBLEN           256
49 #define SMBD_PRINTER           "Postscript"
50 #define SMBD_FN_PREFIX         "cifsprintjob-"
51 #define SMBD_CUPS_SPOOL_DIR    "//var/spool/cups"
52 #define SMBD_CUPS_DOCNAME      "generic_doc"

```

```

54 typedef struct smbd_printjob {
55     pid_t      pj_pid;
56     int        pj_sysjob;
57     int        pj_fd;
58     time_t     pj_start_time;
59     int        pj_status;
60     size_t     pj_size;
61     int        pj_page_count;
62     boolean_t  pj_isspooled;
63     boolean_t  pj_jobnum;
64     char       pj_filename[SMBD_PJOBLEN];
65     char       pj_jobname[SMBD_PJOBLEN];
66     char       pj_username[SMBD_PJOBLEN];
67     char       pj_queue_name[SMBD_PJOBLEN];
68 } smbd_printjob_t;

70 typedef struct smb_cups_ops {
71     void        *cups_hdl;
72     cups_lang_t (*cupsLangDefault)();
73     const char  (*cupsLangEncoding)(cups_lang_t *);
74     void        (*cupsLangFree)(cups_lang_t *);
75     ipp_status_t (*cupsLastError)();
76     int         (*cupsGetDests)(cups_dest_t **);
77     void        (*cupsFreeDests)(int, cups_dest_t *);
78     ipp_t       (*cupsDoFileRequest)(http_t *, ipp_t *,
79     const char *, const char *);
80     ipp_t       (*ippNew)();
81     void        (*ippDelete)();
82     char        (*ippErrorString)();
83     ipp_attribute_t (*ippAddString)();
84     void        (*httpClose)(http_t *);
85     http_t      (*httpConnect)(const char *, int);
86 } smb_cups_ops_t;

88 static uint32_t smbd_cups_jobnum = 1;
89 static smb_cups_ops_t smb_cups;
90 static mutex_t smbd_cups_mutex;

92 static void *smbd_spool_monitor(void *);
93 static smb_cups_ops_t *smbd_cups_ops(void);
94 static void smbd_print_share_comment(smb_share_t *, cups_dest_t *);
95 static void *smbd_share_printers(void *);
96 static void smbd_spool_copyfile(smb_inaddr_t *, char *, char *, char *);

98 extern smbd_t smbd;

100 /*
101  * Start the spool thread.
102  * Returns 0 on success, an error number if thread creation fails.
103  */
104 void
105 smbd_spool_start(void)
106 {
107     pthread_attr_t attr;
108     int rc;

110     if (!smbd_config_getbool(SMB_CI_PRINT_ENABLE))
111         return;

113     (void) pthread_attr_init(&attr);
114     (void) pthread_attr_setdetachstate(&attr, PTHREAD_CREATE_DETACHED);
115     rc = pthread_create(&smbd.s_spool_tid, &attr, smbd_spool_monitor, NULL);
116     (void) pthread_attr_destroy(&attr);

118     if (rc != 0)
119         smbd_log(smbd.s_loghd, LOG_NOTICE,

```

```

120         "failed to start print monitor: %s", strerror(errno));
121     }

123 /*
124  * A single pthread_kill should be sufficient but we include
125  * a couple of retries to avoid implementation idiosyncrasies
126  * around signal delivery.
127  */
128 void
129 smbd_spool_stop(void)
130 {
131     int    i;

133     if (pthread_self() == smbd.s_spool_tid)
134         return;

136     for (i = 0; i < 3 && smbd.s_spool_tid != 0; ++i) {
137         if (pthread_kill(smbd.s_spool_tid, SIGTERM) == ESRCH)
138             break;

140         (void) sleep(1);
141     }
142 }

144 /*
145  * This thread blocks waiting for close print file in the kernel.
146  * It then uses the data returned from the ioctl to copy the spool file
147  * into the cups spooler.
148  *
149  * This mechanism is really only used by Windows Vista and Windows 7.
150  * Other versions of Windows create a zero size file, which is removed
151  * by smbd_spool_copyfile.
152  */
153 /*ARGSUSED*/
154 static void *
155 smbd_spool_monitor(void *arg)
156 {
157     uint32_t    spool_num;
158     char        username[MAXNAMELEN];
159     char        path[MAXPATHLEN];
160     smb_inaddr_t    ipaddr;
161     int         error_retry_cnt = 5;

163     smbd_online_wait("smbd_spool_monitor");

165     spoolss_register_copyfile(smbd_spool_copyfile);

167     while (!smbd.s_shutting_down && (error_retry_cnt > 0)) {
168         errno = 0;

170         if (smb_kmod_get_spool_doc(&spool_num, username,
171             path, &ipaddr) == 0) {
172             smbd_spool_copyfile(&ipaddr,
173                 username, path, SMBD_CUPS_DOCNAME);
174             error_retry_cnt = 5;
175         } else {
176             if (errno == ECANCELED)
177                 break;
178             if ((errno != EINTR) && (errno != EAGAIN))
179                 error_retry_cnt--;
180             (void) sleep(SMB_SPOOL_WAIT);
181         }
182     }

184     spoolss_register_copyfile(NULL);
185     smbd.s_spool_tid = 0;

```

```

186         return (NULL);
187     }

189 /*
190  * All versions of windows use this function to spool files to a printer
191  * via the cups interface
192  */
193 static void
194 smbd_spool_copyfile(smb_inaddr_t *ipaddr, char *username, char *path,
195     char *doc_name)
196 {
197     smb_cups_ops_t    *cups;
198     http_t            *http = NULL;          /* HTTP connection to server */
199     ipp_t              *request = NULL;      /* IPP Request */
200     ipp_t              *response = NULL;     /* IPP Response */
201     cups_lang_t       *language = NULL;    /* Default language */
202     char               uri[HTTP_MAX_URI];   /* printer-uri attribute */
203     char               new_jobname[SMBD_PJOBLEN];
204     smbd_printjob_t    pjob;
205     char               clientname[INET6_ADDRSTRLEN];
206     struct stat        sbuf;
207     int                rc = 1;

209     if (stat(path, &sbuf)) {
210         smb_log(smbd.s_loghd, LOG_INFO, "smbd_spool_copyfile: %s: %s",
211             path, strerror(errno));
212         return;
213     }

215     /*
216     * Remove zero size files and return; these were inadvertently
217     * created by XP or 2000.
218     */
219     if (sbuf.st_size == 0) {
220         if (remove(path) != 0)
221             smb_log(smbd.s_loghd, LOG_INFO,
222                 "smbd_spool_copyfile: cannot remove %s: %s",
223                 path, strerror(errno));
224         return;
225     }

227     if ((cups = smbd_cups_ops()) == NULL)
228         return;

230     if ((http = cups->httpConnect("localhost", 631)) == NULL) {
231         smb_log(smbd.s_loghd, LOG_INFO,
232             "smbd_spool_copyfile: cupsd not running");
233         return;
234     }

236     if ((request = cups->ippNew()) == NULL) {
237         smb_log(smbd.s_loghd, LOG_INFO,
238             "smbd_spool_copyfile: ipp not running");
239         return;
240     }

242     request->request.op.operation_id = IPP_PRINT_JOB;
243     request->request.op.request_id = 1;
244     language = cups->cupsLangDefault();

246     cups->ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_CHARSET,
247         "attributes-charset", NULL, cups->cupsLangEncoding(language));

249     cups->ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_LANGUAGE,
250         "attributes-natural-language", NULL, language->language);

```

```

252     (void) snprintf(uri, sizeof (uri), "ipp://localhost/printers/%s",
253                    SMBD_PRINTER);
254     pjob.pj_pid = pthread_self();
255     pjob.pj_sysjob = 10;
256     (void) strncpy(pjob.pj_filename, path, SMBD_PJOBLEN);
257     pjob.pj_start_time = time(NULL);
258     pjob.pj_status = 2;
259     pjob.pj_size = sbuf.st_blocks * 512;
260     pjob.pj_page_count = 1;
261     pjob.pj_isspoiled = B_TRUE;
262     pjob.pj_jobnum = smbd_cups_jobnum;

264     (void) strncpy(pjob.pj_jobname, doc_name, SMBD_PJOBLEN);
265     (void) strncpy(pjob.pj_username, username, SMBD_PJOBLEN);
266     (void) strncpy(pjob.pj_queue_name, SMBD_CUPS_SPOOL_DIR, SMBD_PJOBLEN);

268     cups->ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_URI,
269                       "printer-uri", NULL, uri);

271     cups->ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_NAME,
272                       "requesting-user-name", NULL, pjob.pj_username);

274     if (smb_inet_ntop(ipaddr, clientname,
275                     SMB_IPSTRLEN(ipaddr->a_family)) == NULL) {
276         smb_log(smbd.s_loghd, LOG_INFO,
277               "smbd_spool_copyfile: %s: unknown client", clientname);
278         goto out;
279     }

281     cups->ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_NAME,
282                       "job-originating-host-name", NULL, clientname);

284     (void) snprintf(new_jobname, SMBD_PJOBLEN, "%s%d",
285                    SMBD_FN_PREFIX, pjob.pj_jobnum);
286     cups->ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_NAME,
287                       "job-name", NULL, new_jobname);

289     (void) snprintf(uri, sizeof (uri) - 1, "/printers/%s", SMBD_PRINTER);

291     response = cups->cupsDoFileRequest(http, request, uri,
292                                       pjob.pj_filename);
293     if (response != NULL) {
294         if (response->request.status.status_code >= IPP_OK_CONFLICT) {
295             smb_log(smbd.s_loghd, LOG_ERR,
296                   "smbd_spool_copyfile: printer %s: %s",
297                   SMBD_PRINTER,
298                   cups->ippErrorString(cups->cupsLastError()));
299         } else {
300             atomic_inc_32(&smbd_cups_jobnum);
301             rc = 0;
302         }
303     } else {
304         smb_log(smbd.s_loghd, LOG_ERR,
305               "smbd_spool_copyfile: unable to print to %s",
306               cups->ippErrorString(cups->cupsLastError()));
307     }

309     if (rc == 0)
310         (void) unlink(pjob.pj_filename);

312 out:
313     if (response)
314         cups->ippDelete(response);

316     if (language)
317         cups->cupsLangFree(language);

```

```

319     if (http)
320         cups->httpClose(http);
321 }

323 int
324 smbd_cups_init(void)
325 {
326     (void) mutex_lock(&smbd_cups_mutex);

328     if (smb_cups.cups_hdl != NULL) {
329         (void) mutex_unlock(&smbd_cups_mutex);
330         return (0);
331     }

333     if ((smb_cups.cups_hdl = dlopen("libcups.so.2", RTLD_NOW)) == NULL) {
334         (void) mutex_unlock(&smbd_cups_mutex);
335         smb_log(smbd.s_loghd, LOG_DEBUG,
336               "smbd_cups_init: cannot open libcups");
337         return (ENOENT);
338     }

340     smbd_cups.cupsLangDefault =
341         (cups_lang_t (*)(void))dlsym(smb_cups.cups_hdl, "cupsLangDefault");
342     smbd_cups.cupsLangEncoding = (const char (*)(cups_lang_t *))
343         dlsym(smb_cups.cups_hdl, "cupsLangEncoding");
344     smbd_cups.cupsDoFileRequest =
345         (ipp_t (*)(http_t *, ipp_t *, const char *, const char *))
346         dlsym(smb_cups.cups_hdl, "cupsDoFileRequest");
347     smbd_cups.cupsLastError = (ipp_status_t (*)(void))
348         dlsym(smb_cups.cups_hdl, "cupsLastError");
349     smbd_cups.cupsLangFree = (void (*)(cups_lang_t *))
350         dlsym(smb_cups.cups_hdl, "cupsLangFree");
351     smbd_cups.cupsGetDests = (int (*)(cups_dest_t **))
352         dlsym(smb_cups.cups_hdl, "cupsGetDests");
353     smbd_cups.cupsFreeDests = (void (*)(int, cups_dest_t *))
354         dlsym(smb_cups.cups_hdl, "cupsFreeDests");

356     smbd_cups.httpClose = (void (*)(http_t *))
357         dlsym(smb_cups.cups_hdl, "httpClose");
358     smbd_cups.httpConnect = (http_t (*)(const char *, int))
359         dlsym(smb_cups.cups_hdl, "httpConnect");

361     smbd_cups.ippNew = (ipp_t (*)(void))dlsym(smb_cups.cups_hdl, "ippNew");
362     smbd_cups.ippDelete = (void (*)(void))dlsym(smb_cups.cups_hdl, "ippDelete");
363     smbd_cups.ippErrorString = (char (*)(void))
364         dlsym(smb_cups.cups_hdl, "ippErrorString");
365     smbd_cups.ippAddString = (ipp_attribute_t (*)(void))
366         dlsym(smb_cups.cups_hdl, "ippAddString");

368     if (smb_cups.cupsLangDefault == NULL ||
369         smbd_cups.cupsLangEncoding == NULL ||
370         smbd_cups.cupsDoFileRequest == NULL ||
371         smbd_cups.cupsLastError == NULL ||
372         smbd_cups.cupsLangFree == NULL ||
373         smbd_cups.cupsGetDests == NULL ||
374         smbd_cups.cupsFreeDests == NULL ||
375         smbd_cups.ippNew == NULL ||
376         smbd_cups.httpClose == NULL ||
377         smbd_cups.httpConnect == NULL ||
378         smbd_cups.ippDelete == NULL ||
379         smbd_cups.ippErrorString == NULL ||
380         smbd_cups.ippAddString == NULL) {
381         (void) dlclose(smb_cups.cups_hdl);
382         smbd_cups.cups_hdl = NULL;
383         (void) mutex_unlock(&smbd_cups_mutex);

```

```

384     smb_log(smbd.s_loghd, LOG_DEBUG,
385             "smbd_cups_init: cannot load libcups");
386     return (ENOENT);
387 }
389 (void) mutex_unlock(&smbd_cups_mutex);
390 return (0);
391 }
393 void
394 smbd_cups_fini(void)
395 {
396     (void) mutex_lock(&smbd_cups_mutex);
398     if (smb_cups.cups_hdl != NULL) {
399         (void) dlclose(smb_cups.cups_hdl);
400         smb_cups.cups_hdl = NULL;
401     }
403     (void) mutex_unlock(&smbd_cups_mutex);
404 }
406 static smb_cups_ops_t *
407 smbd_cups_ops(void)
408 {
409     if (smb_cups.cups_hdl == NULL)
410         return (NULL);
412     return (&smb_cups);
413 }
415 void
416 smbd_load_printers(void)
417 {
418     pthread_t      tid;
419     pthread_attr_t attr;
420     int            rc;
422     if (!smb_config_getbool(SMB_CI_PRINT_ENABLE))
423         return;
425     (void) pthread_attr_init(&attr);
426     (void) pthread_attr_setdetachstate(&attr, PTHREAD_CREATE_DETACHED);
427     rc = pthread_create(&tid, &attr, smbd_share_printers, &tid);
428     (void) pthread_attr_destroy(&attr);
430     if (rc != 0)
431         smb_log(smbd.s_loghd, LOG_NOTICE,
432               "unable to load printer shares: %s", strerror(errno));
433 }
435 /*
436 * All print shares use the path from print$.
437 */
438 /*ARGSUSED*/
439 static void *
440 smbd_share_printers(void *arg)
441 {
442     cups_dest_t    *dests;
443     cups_dest_t    *dest;
444     smb_cups_ops_t *cups;
445     smb_share_t    si;
446     uint32_t       nerr;
447     int            num_dests;
448     int            i;

```

```

450     if (!smb_config_getbool(SMB_CI_PRINT_ENABLE))
451         return (NULL);
453     if ((cups = smbd_cups_ops()) == NULL)
454         return (NULL);
456     if (smb_shr_get(SMB_SHARE_PRINT, &si) != NERR_Success) {
457         smb_log(smbd.s_loghd, LOG_DEBUG,
458               "smbd_share_printers unable to load %s", SMB_SHARE_PRINT);
459         return (NULL);
460     }
462     num_dests = cups->cupsGetDests(&dests);
464     for (i = num_dests, dest = dests; i > 0; i--, dest++) {
465         if (dest->instance != NULL)
466             continue;
468         (void) strncpy(si.shr_name, dest->name, MAXPATHLEN);
469         smbd_print_share_comment(&si, dest);
470         si.shr_type = STYPE_PRINTQ;
472         nerr = smb_shr_add(&si);
473         if (nerr == NERR_Success || nerr == NERR_DuplicateShare)
474             smb_log(smbd.s_loghd, LOG_DEBUG,
475                   "shared printer: %s", si.shr_name);
476         else
477             smb_log(smbd.s_loghd, LOG_DEBUG,
478                   "smbd_share_printers: unable to add share %s: %u",
479                   si.shr_name, nerr);
480     }
482     cups->cupsFreeDests(num_dests, dests);
483     return (NULL);
484 }
486 static void
487 smbd_print_share_comment(smb_share_t *si, cups_dest_t *dest)
488 {
489     cups_option_t  *options;
490     char            *comment;
491     char            *name;
492     char            *value;
493     int            i;
495     comment = "Print Share";
497     if ((options = dest->options) == NULL) {
498         (void) strncpy(si->shr_cmnt, comment, SMB_SHARE_CMNT_MAX);
499         return;
500     }
502     for (i = 0; i < dest->num_options; ++i) {
503         name = options[i].name;
504         value = options[i].value;
506         if (name == NULL || value == NULL ||
507             *name == '\0' || *value == '\0')
508             continue;
510         if (strcascmp(name, "printer-info") == 0) {
511             comment = value;
512             break;
513         }
514     }

```

```
516     (void) strcpy(si->shr_cmnt, comment, SMB_SHARE_CMNT_MAX);
517 }

519 #else /* HAVE_CUPS */

521 /*
522  * If not HAVE_CUPS, just provide a few "stubs".
523  */

525 int
526 smbd_cups_init(void)
527 {
528     return (ENOENT);
529 }

531 void
532 smbd_cups_fini(void)
533 {
534 }

536 void
537 smbd_load_printers(void)
538 {
539 }

541 void
542 smbd_spool_init(void)
543 {
544 }

546 void
547 smbd_spool_fini(void)
548 {
549 }

551 void
552 smbd_spool_start(void)
553 {
554 }

556 void
557 smbd_spool_stop(void)
558 {
559 }

561 #endif /* HAVE_CUPS */
51 #endif
```

```

*****
2440 Thu Feb 13 22:45:19 2014
new/usr/src/cmd/ssh/libssh/Makefile.com
don't pass in lint libraries which no longer exist
*****
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 2009 Sun Microsystems, Inc. All rights reserved.
22 # Use is subject to license terms.
23 #

25 LIBRARY =      libssh.a
26 VERS =        .1

28 OBJECTS =      \
29                addrmatch.o \
30                atomicio.o \
31                authfd.o \
32                authfile.o \
33                bufaux.o \
34                buffer.o \
35                canohost.o \
36                channels.o \
37                cipher.o \
38                cipher-ctr.o \
39                compat.o \
40                compress.o \
41                crc32.o \
42                deattack.o \
43                dh.o \
44                dispatch.o \
45                engine.o \
46                entropy.o \
47                fatal.o \
48                glln.o \
49                hostfile.o \
50                key.o \
51                kex.o \
52                kexdh.o \
53                kexdhc.o \
54                kexdhs.o \
55                kexgex.o \
56                kexgexc.o \
57                kexgexs.o \
58                kexgssc.o \
59                kexgsss.o \
60                log.o \
61                mac.o \

```

```

62                match.o \
63                misc.o \
64                mpaux.o \
65                msg.o \
66                nchan.o \
67                packet.o \
68                progressmeter.o \
69                proxy-io.o \
70                radix.o \
71                readconf.o \
72                readpass.o \
73                rsa.o \
74                sftp-common.o \
75                ssh-dss.o \
76                ssh-gss.o \
77                ssh-rsa.o \
78                tildexpand.o \
79                ttymodes.o \
80                uidswap.o \
81                uuencode.o \
82                xlist.o \
83                xmalloc.o

85 include $(SRC)/lib/Makefile.lib

87 BUILD.AR =     $(RM) $@ ; $(AR) $(ARFLAGS) $@ $(AROBJS)

89 SRCDIR =       ../common
90 SRCS =          $(OBJECTS:%.o=../common/%.c)

92 LIBS =         $(LIBRARY) $(LINTLIB)

94 # Define LDLIBS conditionally for lintcheck, rather than in general, since
95 # we're building an archive library which itself links to nothing, we just
96 # want lint to know about these libraries.
97 lintcheck :=   LDLIBS += -lsocket -lnsl -lc
98 lintcheck :=   LDLIBS += -lcrypto -lz -lsocket -lnsl -lc
99 $(LINTLIB) :=   SRCS = $(SRCDIR)/$(LINTSRC)

100 POFILE_DIR =   ../..

102 .KEEP_STATE:

104 all:           $(LIBS)

106 # lint requires the (not installed) lint library
107 lint:          $(LINTLIB) .WAIT lintcheck

109 include $(SRC)/lib/Makefile.targ

111 objs%.o:       $(SRCDIR)/%.c
112                $(COMPILE.c) -o $@ $<
113                $(POST_PROCESS_O)

115 include ../../Makefile.ssh-common
116 include ../../Makefile.msg.targ

```

new/usr/src/cmd/ssh/sftp-server/Makefile

1

```
*****
1595 Thu Feb 13 22:45:19 2014
new/usr/src/cmd/ssh/sftp-server/Makefile
don't pass in lint libraries which no longer exist
*****
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 2009 Sun Microsystems, Inc. All rights reserved.
22 # Use is subject to license terms.
23 #
24 # cmd/ssh/sftp-server/Makefile

26 PROG =          sftp-server

28 OBJS =          sftp-server.o sftp-server-main.o
29 SRCS =          $(OBJS:.o=.c)

31 include .././Makefile.cmd
32 include ../Makefile.ssh-common

34 LDLIBS +=       $(SSH_COMMON_LDLIBS) -lsocket

36 # libcrypto has no lint library, so we can only use it when building
37 $(PROG) :=      LDLIBS += -lcrypto
34 LDLIBS +=       $(SSH_COMMON_LDLIBS) -lsocket -lcrypto

39 POFILE_DIR =    ..

41 .KEEP_STATE:

43 .PARALLEL:      $(OBJS)

45 all:            $(PROG)

47 $(PROG):        $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/
48                 $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
49                 $(POST_PROCESS)

51 install:        all $(ROOTLIBSSHPROG) $(ROOTLIBSSH)

53 clean:
54                 $(RM) -f $(OBJS) $(PROG)

56 lint:           lint_SRCS

58 include ../Makefile.msg.targ
59 include .././Makefile.targ
```

new/usr/src/cmd/ssh/sftp/Makefile

1

```
*****
1579 Thu Feb 13 22:45:19 2014
new/usr/src/cmd/ssh/sftp/Makefile
don't pass in lint libraries which no longer exist
*****
```

```
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 2009 Sun Microsystems, Inc. All rights reserved.
22 # Use is subject to license terms.
23 #
24 # cmd/ssh/sftp/Makefile

26 PROG =          sftp

28 OBJS =          \
29                sftp.o \
30                sftp-client.o \
31                sftp-glob.o

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

35 include ../../Makefile.cmd
36 include ../../Makefile.ssh-common

38 LDLIBS +=       $(SSH_COMMON_LDLIBS) -lsocket -ltecla
39 # libcrypto has no lint library, so we can only use it when building
40 $(PROG) := LDLIBS += -lcrypto
38 LDLIBS +=       $(SSH_COMMON_LDLIBS) -lsocket -lcrypto -ltecla

42 POFILE_DIR =    ..

44 .KEEP_STATE:

46 .PARALLEL:     $(OBJJS)

48 all: $(PROG)

50 $(PROG):        $(OBJJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/
51                $(LINK.c) $(OBJJS) -o $@ $(LDLIBS) $(DYNFLAGS)
52                $(POST_PROCESS)

54 install:       all $(ROOTPROG)

56 clean:
57               $(RM) -f $(OBJJS) $(PROG)

59 lint:          lint_SRCS
```

new/usr/src/cmd/ssh/sftp/Makefile

2

```
61 include ../Makefile.msg.targ
62 include ../../Makefile.targ
```



new/usr/src/cmd/ssh/ssh-add/Makefile

1

```
*****
1542 Thu Feb 13 22:45:20 2014
new/usr/src/cmd/ssh/ssh-add/Makefile
don't pass in lint libraries which no longer exist
*****
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/ssh/ssh-add/Makefile

26 PROG= ssh-add

28 OBJS   =      \
29         ssh-add.o
30 SRCS   = $(OBJS:.o=.c)

32 include .././Makefile.cmd
33 include .././Makefile.ssh-common

35 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket

37 # libcrypto has no lint library, so we can only use it when building
38 $(PROG) := LDLIBS += -lcrypto
35 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket -lcrypto

40 POFILE_DIR= ..

42 .KEEP_STATE:

44 .PARALLEL: $(OBJS)

46 all: $(PROG)

48 $(PROG): $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/libopen
49          $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
50          $(POST_PROCESS)

52 install: all $(ROOTPROG)

54 clean:
55         $(RM) -f $(OBJS) $(PROG)

57 lint:  lint_SRCS

59 include .././Makefile.msg.targ
60 include .././Makefile.targ
```

new/usr/src/cmd/ssh/ssh-agent/Makefile

1

\*\*\*\*\*

1548 Thu Feb 13 22:45:20 2014

new/usr/src/cmd/ssh/ssh-agent/Makefile

don't pass in lint libraries which no longer exist

\*\*\*\*\*

```
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/ssh/ssh-agent/Makefile

26 PROG= ssh-agent

28 OBJS   =      \
29         ssh-agent.o
30 SRCS   = $(OBJS:.o=.c)

32 include .././Makefile.cmd
33 include ../Makefile.ssh-common

35 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket

37 # libcrypto has no lint library, so we can only use it when building
38 $(PROG) := LDLIBS += -lcrypto
35 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket -lcrypto

40 POFILE_DIR= ..

42 .KEEP_STATE:

44 .PARALLEL: $(OBJS)

46 all: $(PROG)

48 $(PROG): $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/libopen
49         $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
50         $(POST_PROCESS)

52 install: all $(ROOTPROG)

54 clean:
55         $(RM) -f $(OBJS) $(PROG)

57 lint:  lint_SRCS

59 include ../Makefile.msg.targ
60 include .././Makefile.targ
```

new/usr/src/cmd/ssh/ssh-keygen/Makefile

1

```
*****
1551 Thu Feb 13 22:45:21 2014
new/usr/src/cmd/ssh/ssh-keygen/Makefile
don't pass in lint libraries which no longer exist
*****
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/ssh/ssh-keygen/Makefile

26 PROG= ssh-keygen

28 OBJS   =      \
29         ssh-keygen.o
30 SRCS   = $(OBJS:.o=.c)

32 include .././Makefile.cmd
33 include ../Makefile.ssh-common

35 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket

37 # libcrypto has no lint library, so we can only use it when building
38 $(PROG) := LDLIBS += -lcrypto
35 LDLIBS += $(SSH_COMMON_LDLIBS) -lcrypto -lsocket

40 POFILE_DIR= ..

42 .KEEP_STATE:

44 .PARALLEL: $(OBJS)

46 all: $(PROG)

48 $(PROG): $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/libopen
49         $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
50         $(POST_PROCESS)

52 install: all $(ROOTPROG)

54 clean:
55         $(RM) -f $(OBJS) $(PROG)

57 lint: lint_SRCS

59 include ../Makefile.msg.targ
60 include .././Makefile.targ
```

new/usr/src/cmd/ssh/ssh-keyscan/Makefile

1

\*\*\*\*\*

1574 Thu Feb 13 22:45:21 2014

new/usr/src/cmd/ssh/ssh-keyscan/Makefile

don't pass in lint libraries which no longer exist

\*\*\*\*\*

```
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/ssh/ssh-keyscan/Makefile

26 PROG= ssh-keyscan

28 OBJS   =      \
29         ssh-keyscan.o
30 SRCS   = $(OBJS:.o=.c)

32 include ../../Makefile.cmd
33 include ../../Makefile.ssh-common

35 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket -lnsl

37 # libcrypto and libz have no lint library, so we can only use it when building
38 $(PROG) := LDLIBS += -lcrypto -lz
35 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket -lnsl -lz -lcrypto

40 POFILE_DIR= ..

42 .KEEP_STATE:

44 .PARALLEL: $(OBJS)

46 all: $(PROG)

48 $(PROG): $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/libopen
49         $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
50         $(POST_PROCESS)

52 clean:
53         $(RM) -f $(OBJS) $(PROG)

55 lint: lint_SRCS

57 include ../../Makefile.msg.targ
58 include ../../Makefile.targ

60 install: all $(ROOTPROG)
```

```

*****
1694 Thu Feb 13 22:45:22 2014
new/usr/src/cmd/ssh/ssh/Makefile
don't pass in lint libraries which no longer exist
*****
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/ssh/ssh/Makefile

26 PROG= ssh

28 OBJS    = ssh.o \
29          sshconnect.o \
30          sshconnect1.o \
31          sshconnect2.o \
32          sstty.o \
33          clientloop.o \
34          gss-clnt.o
35 SRCS    = $(OBJS:.o=.c)

37 include ../../Makefile.cmd
38 include ../Makefile.ssh-common

40 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket \
41          -lnsl \
42          -lz \
43          -lcrypto \
44          -lgss

44 # libcrypto and libz have no lint libraries, so we can only use them when
45 # building
46 $(PROG) := LDLIBS += -lcrypto -lz

48 #endif /* ! codereview */
49 PROFILE_DIR= ..

51 .KEEP_STATE:

53 .PARALLEL: $(OBJS)

55 all: $(PROG)

57 $(PROG): $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/libopen
58          $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
59          $(POST_PROCESS)

```

```

61 install: all $(ROOTPROG)

63 clean:
64      $(RM) -f $(OBJS) $(PROG)

66 lint:  lint_SRCS

68 include ../Makefile.msg.targ

70 XGETFLAGS += --keyword=log
71 include ../../Makefile.targ

```

new/usr/src/lib/libkmf/plugins/kmf\_openssl/Makefile.com

1

\*\*\*\*\*  
1939 Thu Feb 13 22:45:22 2014

new/usr/src/lib/libkmf/plugins/kmf\_openssl/Makefile.com  
don't pass in lint libraries which no longer exist

\*\*\*\*\*

```
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 2009 Sun Microsystems, Inc. All rights reserved.
22 # Use is subject to license terms.
23 #
24 # Makefile for KMF Plugins
25 #

27 LIBRARY=      kmf_openssl.a
28 VERS=        .1

30 OBJECTS=      openssl_spi.o

32 include $(SRC)/lib/Makefile.lib

34 LIBLINKS=     $(DYNLIB:.so.1=.so)
35 KMFINC=       -I../../include -I../../ber_der/inc

37 BERLIB=       -lkmf -lkmfberder
38 BERLIB64=     $(BERLIB)

40 OPENSLLIBS=   $(BERLIB) -lcrypto -lcryptoutil -lc
41 OPENSLLIBS64= $(BERLIB64) -lcrypto -lcryptoutil -lc

43 LINTSSLLIBS   = $(BERLIB) -lcryptoutil -lc
44 LINTSSLLIBS64 = $(BERLIB64) -lcryptoutil -lc
43 LINTSSLLIBS   = $(BERLIB) -lcrypto -lcryptoutil -lc
44 LINTSSLLIBS64 = $(BERLIB64) -lcrypto -lcryptoutil -lc

46 SRCDIR=      ../common
47 INCDIR=      ../../include

49 CFLAGS       += $(CCVERBOSE)
50 CPPFLAGS     += -D_REENTRANT $(KMFINC) \
51              -I$(INCDIR) -I$(ADJUNCT_PROTO)/usr/include/libxml2

53 CERRWARN     += -_gcc=-Wno-unused-label
54 CERRWARN     += -_gcc=-Wno-unused-value
55 CERRWARN     += -_gcc=-Wno-uninitialized

57 PICS=        $(OBJECTS:%=pics/%)

59 lint:=       OPENSLLIBS= $(LINTSSLLIBS)
```

new/usr/src/lib/libkmf/plugins/kmf\_openssl/Makefile.com

2

```
60 lint:=       OPENSLLIBS64= $(LINTSSLLIBS64)

62 LDLIBS32     += $(OPENSLLIBS)

64 ROOTLIBDIR=  $(ROOTFS_LIBDIR)/crypto
65 ROOTLIBDIR64= $(ROOTFS_LIBDIR)/crypto/$(MACH64)

67 .KEEP_STATE:

69 LIBS        = $(DYNLIB)
70 all:        $(DYNLIB) $(LINTLIB)

72 lint: lintcheck

74 FRC:

76 include $(SRC)/lib/Makefile.targ
```

```

*****
133359 Thu Feb 13 22:45:23 2014
new/usr/src/lib/libkmf/plugins/kmf_openssl/common/openssl_spi.c
(void) an sk_set via a macro which upsets lint. If the macro is
complicated, this might need to be /* LINTED */ instead
remove lint suppression made unnecessary by the openssl upgrade
*****
_____unchanged_portion_omitted_____

2483 /* ocsf_find_signer_sk() is copied from openssl source */
2484 static X509 *ocsf_find_signer_sk(STACK_OF(X509) *certs, OCSP_RESPID *id)
2485 {
2486     int i;
2487     unsigned char tmphash[SHA_DIGEST_LENGTH], *keyhash;

2489     /* Easy if lookup by name */
2490     if (id->type == V_OCSP_RESPID_NAME)
2491         return (X509_find_by_subject(certs, id->value.byName));

2493     /* Lookup by key hash */

2495     /* If key hash isn't SHA1 length then forget it */
2496     if (id->value.byKey->length != SHA_DIGEST_LENGTH)
2497         return (NULL);

2499     keyhash = id->value.byKey->data;
2500     /* Calculate hash of each key and compare */
2501     for (i = 0; i < sk_X509_num(certs); i++) {
2502         /* LINTED E_BAD_PTR_CAST_ALIGN */
2503         X509 *x = sk_X509_value(certs, i);
2504         /* Use pubkey_digest to get the key ID value */
2505         (void) X509_pubkey_digest(x, EVP_sha1(), tmphash, NULL);
2506         if (!memcmp(keyhash, tmphash, SHA_DIGEST_LENGTH))
2507             return (x);
2508     }
2509     return (NULL);

_____unchanged_portion_omitted_____

3595 /*
3596 * Helper function to extract keys and certificates from
3597 * a single PEM file. Typically the file should contain a
3598 * private key and an associated public key wrapped in an x509 cert.
3599 * However, the file may be just a list of X509 certs with no keys.
3600 */
3601 static KMF_RETURN
3602 extract_pem(KMF_HANDLE *kmfh,
3603            char *issuer, char *subject, KMF_BIGNT *serial,
3604            char *filename, CK_UTF8CHAR *pin,
3605            CK_ULONG pinlen, EVP_PKEY **priv_key, KMF_DATA **certs,
3606            int *numcerts)
3607 /* ARGSUSED6 */
3608 {
3609     KMF_RETURN rv = KMF_OK;
3610     FILE *fp;
3611     STACK_OF(X509_INFO) *x509_info_stack = NULL;
3612     int i, ncerts = 0, matchcerts = 0;
3613     EVP_PKEY *pkey = NULL;
3614     X509_INFO *info;
3615     X509 *x;
3616     X509_INFO **cert_infos = NULL;
3617     KMF_DATA *certlist = NULL;

3619     if (priv_key)
3620         *priv_key = NULL;
3621     if (certs)

```

```

3622         *certs = NULL;
3623         fp = fopen(filename, "r");
3624         if (fp == NULL)
3625             return (KMF_ERR_OPEN_FILE);

3627         x509_info_stack = PEM_X509_INFO_read(fp, NULL, NULL, pin);
3628         if (x509_info_stack == NULL) {
3629             (void) fclose(fp);
3630             return (KMF_ERR_ENCODING);
3631         }
3632         cert_infos = (X509_INFO **)malloc(sk_X509_INFO_num(x509_info_stack) *
3633             sizeof (X509_INFO *));
3634         if (cert_infos == NULL) {
3635             (void) fclose(fp);
3636             rv = KMF_ERR_MEMORY;
3637             goto err;
3638         }

3640         for (i = 0; i < sk_X509_INFO_num(x509_info_stack); i++) {
3642             /* LINTED E_BAD_PTR_CAST_ALIGN */
3641             cert_infos[ncerts] = sk_X509_INFO_value(x509_info_stack, i);
3642             ncerts++;
3643         }

3645         if (ncerts == 0) {
3646             (void) fclose(fp);
3647             rv = KMF_ERR_CERT_NOT_FOUND;
3648             goto err;
3649         }

3651         if (priv_key != NULL) {
3652             rewind(fp);
3653             pkey = PEM_read_PrivateKey(fp, NULL, NULL, pin);
3654         }
3655         (void) fclose(fp);

3657         x = cert_infos[ncerts - 1]->x509;
3658         /*
3659          * Make sure the private key matches the last cert in the file.
3660          */
3661         if (pkey != NULL && !X509_check_private_key(x, pkey)) {
3662             EVP_PKEY_free(pkey);
3663             rv = KMF_ERR_KEY_MISMATCH;
3664             goto err;
3665         }

3667         certlist = (KMF_DATA *)calloc(ncerts, sizeof (KMF_DATA));
3668         if (certlist == NULL) {
3669             if (pkey != NULL)
3670                 EVP_PKEY_free(pkey);
3671             rv = KMF_ERR_MEMORY;
3672             goto err;
3673         }

3675         /*
3676          * Convert all of the certs to DER format.
3677          */
3678         matchcerts = 0;
3679         for (i = 0; rv == KMF_OK && certs != NULL && i < ncerts; i++) {
3680             boolean_t match = FALSE;
3681             info = cert_infos[ncerts - 1 - i];

3683             rv = check_cert(info->x509, issuer, subject, serial, &match);
3684             if (rv != KMF_OK || match != TRUE) {
3685                 rv = KMF_OK;
3686                 continue;

```

```

3687     }
3689     rv = ssl_cert2KMFDATA(kmfh, info->x509,
3690                          &certlist[matchcerts++]);
3692     if (rv != KMF_OK) {
3693         int j;
3694         for (j = 0; j < matchcerts; j++)
3695             kmf_free_data(&certlist[j]);
3696         free(certlist);
3697         certlist = NULL;
3698         ncerts = matchcerts = 0;
3699     }
3700 }
3702 if (numcerts != NULL)
3703     *numcerts = matchcerts;
3705 if (certs != NULL)
3706     *certs = certlist;
3707 else if (certlist != NULL) {
3708     for (i = 0; i < ncerts; i++)
3709         kmf_free_data(&certlist[i]);
3710     free(certlist);
3711     certlist = NULL;
3712 }
3714 if (priv_key == NULL && pkey != NULL)
3715     EVP_PKEY_free(pkey);
3716 else if (priv_key != NULL && pkey != NULL)
3717     *priv_key = pkey;
3719 err:
3720 /* Cleanup the stack of X509 info records */
3721 for (i = 0; i < sk_X509_INFO_num(x509_info_stack); i++) {
3722     /* LINTED E_BAD_PTR_CAST_ALIGN */
3723     info = (X509_INFO *)sk_X509_INFO_value(x509_info_stack, i);
3724     X509_INFO_free(info);
3725 }
3726 if (x509_info_stack)
3727     sk_X509_INFO_free(x509_info_stack);
3728 if (cert_infos != NULL)
3729     free(cert_infos);
3731 return (rv);
3732 }
3734 static KMF_RETURN
3735 openssl_parse_bags(STACK_OF(PKCS12_SAFEBAG) *bags, char *pin,
3736                  STACK_OF(EVP_PKEY) *keys, STACK_OF(X509) *certs)
3737 {
3738     KMF_RETURN ret;
3739     int i;
3741     for (i = 0; i < sk_PKCS12_SAFEBAG_num(bags); i++) {
3742         /* LINTED E_BAD_PTR_CAST_ALIGN */
3743         PKCS12_SAFEBAG *bag = sk_PKCS12_SAFEBAG_value(bags, i);
3744         ret = openssl_parse_bag(bag, pin, (pin ? strlen(pin) : 0),
3745                                keys, certs);
3746     }
3747     if (ret != KMF_OK)
3748         return (ret);
3749 }
3750 return (ret);

```

```

3751 }
3753 static KMF_RETURN
3754 set_pkey_attr(EVP_PKEY *pkey, ASN1_TYPE *attrib, int nid)
3755 {
3756     X509_ATTRIBUTE *attr = NULL;
3758     if (pkey == NULL || attrib == NULL)
3759         return (KMF_ERR_BAD_PARAMETER);
3761     if (pkey->attributes == NULL) {
3762         pkey->attributes = sk_X509_ATTRIBUTE_new_null();
3763         if (pkey->attributes == NULL)
3764             return (KMF_ERR_MEMORY);
3765     }
3766     attr = X509_ATTRIBUTE_create(nid, attrib->type, attrib->value.ptr);
3767     if (attr != NULL) {
3768         int i;
3769         X509_ATTRIBUTE *a;
3770         for (i = 0;
3771              i < sk_X509_ATTRIBUTE_num(pkey->attributes); i++) {
3772             /* LINTED E_BAD_PTR_CASE_ALIGN */
3773             a = sk_X509_ATTRIBUTE_value(pkey->attributes, i);
3774             if (OBJ_obj2nid(a->object) == nid) {
3775                 X509_ATTRIBUTE_free(a);
3776                 (void) sk_X509_ATTRIBUTE_set(pkey->attributes,
3777                                             /* LINTED E_BAD_PTR_CAST_ALIGN */
3778                                             sk_X509_ATTRIBUTE_set(pkey->attributes,
3779                                                                     i, attr);
3780                 return (KMF_OK);
3781             }
3782         }
3783         if (sk_X509_ATTRIBUTE_push(pkey->attributes, attr) == NULL) {
3784             X509_ATTRIBUTE_free(attr);
3785             return (KMF_ERR_MEMORY);
3786         }
3788     }
3789     return (KMF_OK);
3790 }
3791
3792 unchanged_portion_omitted
3793
3794 static KMF_RETURN
3795 openssl_pkcs12_parse(PKCS12 *p12, char *pin,
3796                    STACK_OF(EVP_PKEY) *keys,
3797                    STACK_OF(X509) *certs,
3798                    STACK_OF(X509) *ca)
3799 /* ARGSUSED3 */
3800 {
3801     KMF_RETURN ret = KMF_OK;
3802     STACK_OF(PKCS7) *asafes = NULL;
3803     STACK_OF(PKCS12_SAFEBAG) *bags = NULL;
3804     int i, bagnid;
3805     PKCS7 *p7;
3807     if (p12 == NULL || (keys == NULL && certs == NULL))
3808         return (KMF_ERR_BAD_PARAMETER);
3810     if (pin == NULL || *pin == NULL) {
3811         if (PKCS12_verify_mac(p12, NULL, 0)) {
3812             pin = NULL;
3813         } else if (PKCS12_verify_mac(p12, "", 0)) {
3814             pin = "";
3815         } else {

```



```

3941         return (KMF_ERR_AUTH_FAILED);
3942     }
3943 } else if (!PKCS12_verify_mac(pl2, pin, -1)) {
3944     return (KMF_ERR_AUTH_FAILED);
3945 }
3947 if ((asafes = PKCS12_unpack_authsafes(pl2)) == NULL)
3948     return (KMF_ERR_PKCS12_FORMAT);
3950 for (i = 0; ret == KMF_OK && i < sk_PKCS7_num(asafes); i++) {
3951     bags = NULL;
3952     /* LINTED E_BAD_PTR_CAST_ALIGN */
3953     p7 = sk_PKCS7_value(asafes, i);
3954     bagnid = OBJ_obj2nid(p7->type);
3955     if (bagnid == NID_pkcs7_data) {
3956         bags = PKCS12_unpack_p7data(p7);
3957     } else if (bagnid == NID_pkcs7_encrypted) {
3958         bags = PKCS12_unpack_p7encdata(p7, pin,
3959             (pin ? strlen(pin) : 0));
3960     } else {
3961         continue;
3962     }
3963     if (bags == NULL) {
3964         ret = KMF_ERR_PKCS12_FORMAT;
3965         goto out;
3966     }
3968     if (openssl_parse_bags(bags, pin, keys, certs) != KMF_OK)
3969         ret = KMF_ERR_PKCS12_FORMAT;
3971     sk_PKCS12_SAFEBAG_pop_free(bags, PKCS12_SAFEBAG_free);
3972 }
3973 out:
3974 if (asafes != NULL)
3975     sk_PKCS7_pop_free(asafes, PKCS7_free);
3977 return (ret);
3978 }

```

unchanged portion omitted

```

4219 static X509_ATTRIBUTE *
4220 find_attr(STACK_OF(X509_ATTRIBUTE) *attrs, int nid)
4221 {
4222     X509_ATTRIBUTE *a;
4223     int i;
4225     if (attrs == NULL)
4226         return (NULL);
4228     for (i = 0; i < sk_X509_ATTRIBUTE_num(attrs); i++) {
4229         /* LINTED E_BAD_PTR_CAST_ALIGN */
4230         a = sk_X509_ATTRIBUTE_value(attrs, i);
4231         if (OBJ_obj2nid(a->object) == nid)
4232             return (a);
4233     }
4234     return (NULL);
4236 static KMF_RETURN
4237 convertToRawKey(EVP_PKEY *pkey, KMF_RAW_KEY_DATA *key)
4238 {
4239     KMF_RETURN rv = KMF_OK;
4240     X509_ATTRIBUTE *attr;
4242     if (pkey == NULL || key == NULL)

```

```

4243         return (KMF_ERR_BAD_PARAMETER);
4244     /* Convert SSL key to raw key */
4245     switch (pkey->type) {
4246     case EVP_PKEY_RSA:
4247         rv = exportRawRSAKey(EVP_PKEY_get1_RSA(pkey),
4248             key);
4249         if (rv != KMF_OK)
4250             return (rv);
4251         break;
4252     case EVP_PKEY_DSA:
4253         rv = exportRawDSAKey(EVP_PKEY_get1_DSA(pkey),
4254             key);
4255         if (rv != KMF_OK)
4256             return (rv);
4257         break;
4258     default:
4259         return (KMF_ERR_BAD_PARAMETER);
4260     }
4261     /*
4262     * If friendlyName, add it to record.
4263     */
4264     attr = find_attr(pkey->attributes, NID_friendlyName);
4265     if (attr != NULL) {
4266         ASN1_TYPE *ty = NULL;
4267         int numattr = sk_ASN1_TYPE_num(attr->value.set);
4268         if (attr->single == 0 && numattr > 0) {
4269             /* LINTED E_BAD_PTR_CAST_ALIGN */
4270             ty = sk_ASN1_TYPE_value(attr->value.set, 0);
4271         }
4272         if (ty != NULL) {
4273             #if OPENSSSL_VERSION_NUMBER < 0x10000000L
4274             key->label = uni2asc(ty->value.bmpstring->data,
4275                 ty->value.bmpstring->length);
4276             #else
4277             key->label = OPENSSSL_uni2asc(ty->value.bmpstring->data,
4278                 ty->value.bmpstring->length);
4279             #endif
4280         } else {
4281             key->label = NULL;
4282         }
4284     /*
4285     * If KeyID, add it to record as a KMF_DATA object.
4286     */
4287     attr = find_attr(pkey->attributes, NID_localKeyID);
4288     if (attr != NULL) {
4289         ASN1_TYPE *ty = NULL;
4290         int numattr = sk_ASN1_TYPE_num(attr->value.set);
4291         if (attr->single == 0 && numattr > 0) {
4292             /* LINTED E_BAD_PTR_CAST_ALIGN */
4293             ty = sk_ASN1_TYPE_value(attr->value.set, 0);
4294         }
4295         key->id.Data = (uchar_t *)malloc(
4296             ty->value.octet_string->length);
4297         if (key->id.Data == NULL)
4298             return (KMF_ERR_MEMORY);
4299         (void) memcpy(key->id.Data, ty->value.octet_string->data,
4300             ty->value.octet_string->length);
4301         key->id.Length = ty->value.octet_string->length;
4302     } else {
4303         (void) memset(&key->id, 0, sizeof (KMF_DATA));
4304     }
4305     return (rv);
4306 }

```

```

4308 static KMF_RETURN
4309 convertPK12Objects(
4310     KMF_HANDLE *kmfh,
4311     STACK_OF(EVP_PKEY) *sslkeys,
4312     STACK_OF(X509) *sslcert,
4313     STACK_OF(X509) *sslcacerts,
4314     KMF_RAW_KEY_DATA **keylist, int *nkeys,
4315     KMF_X509_DER_CERT **certlist, int *ncerts)
4316 {
4317     KMF_RETURN rv = KMF_OK;
4318     KMF_RAW_KEY_DATA key;
4319     int i;

4321     for (i = 0; sslkeys != NULL && i < sk_EVP_PKEY_num(sslkeys); i++) {
4322         /* LINTED E_BAD_PTR_CAST_ALIGN */
4323         EVP_PKEY *pkey = sk_EVP_PKEY_value(sslkeys, i);
4324         rv = convertToRawKey(pkey, &key);
4325         if (rv == KMF_OK)
4326             rv = add_key_to_list(keylist, &key, nkeys);

4327         if (rv != KMF_OK)
4328             return (rv);
4329     }

4331     /* Now add the certificate to the certlist */
4332     for (i = 0; sslcert != NULL && i < sk_X509_num(sslcert); i++) {
4333         /* LINTED E_BAD_PTR_CAST_ALIGN */
4334         X509 *cert = sk_X509_value(sslcert, i);
4335         rv = add_cert_to_list(kmfh, cert, certlist, ncerts);
4336         if (rv != KMF_OK)
4337             return (rv);
4338     }

4339     /* Also add any included CA certs to the list */
4340     for (i = 0; sslcacerts != NULL && i < sk_X509_num(sslcacerts); i++) {
4341         X509 *c;
4342         /*
4343          * sk_X509_value() is macro that embeds a cast to (X509 *).
4344          * Here it translates into ((X509 *)sk_value((ca), (i))).
4345          * Lint is complaining about the embedded casting, and
4346          * to fix it, you need to fix openssl header files.
4347          */
4348         /* LINTED E_BAD_PTR_CAST_ALIGN */
4349         c = sk_X509_value(sslcacerts, i);

4350         /* Now add the ca cert to the certlist */
4351         rv = add_cert_to_list(kmfh, c, certlist, ncerts);
4352         if (rv != KMF_OK)
4353             return (rv);
4354     }
4355     return (rv);
4356 }

```

unchanged portion omitted

```

5309 KMF_RETURN
5310 OpenSSL_FindCertInCRL(KMF_HANDLE_T handle, int numattr, KMF_ATTRIBUTE *attrlist)
5311 {
5312     KMF_RETURN ret = KMF_OK;
5313     KMF_HANDLE *kmfh = (KMF_HANDLE *)handle;
5314     KMF_ENCODE_FORMAT format;
5315     BIO *in = NULL;
5316     X509 *xcert = NULL;
5317     X509_CRL *xcrl = NULL;
5318     STACK_OF(X509_REVOKED) *revoke_stack = NULL;
5319     X509_REVOKED *revoke;

```

```

5320     int i;
5321     char *crlfilename, *crlfile, *dirpath, *certfile;

5323     if (numattr == 0 || attrlist == NULL) {
5324         return (KMF_ERR_BAD_PARAMETER);
5325     }

5327     crlfilename = kmf_get_attr_ptr(KMF_CRL_FILENAME_ATTR,
5328                                   attrlist, numattr);

5330     if (crlfilename == NULL)
5331         return (KMF_ERR_BAD_CRLFILE);

5333     certfile = kmf_get_attr_ptr(KMF_CERT_FILENAME_ATTR, attrlist, numattr);
5334     if (certfile == NULL)
5335         return (KMF_ERR_BAD_CRLFILE);

5337     dirpath = kmf_get_attr_ptr(KMF_DIRPATH_ATTR, attrlist, numattr);

5339     crlfile = get_fullpath(dirpath, crlfilename);

5341     if (crlfile == NULL)
5342         return (KMF_ERR_BAD_CRLFILE);

5344     if (isdir(crlfile)) {
5345         ret = KMF_ERR_BAD_CRLFILE;
5346         goto end;
5347     }

5349     ret = kmf_is_crl_file(handle, crlfile, &format);
5350     if (ret != KMF_OK)
5351         goto end;

5353     /* Read the CRL file and load it into a X509_CRL structure */
5354     in = BIO_new_file(crlfilename, "rb");
5355     if (in == NULL) {
5356         SET_ERROR(kmfh, ERR_get_error());
5357         ret = KMF_ERR_OPEN_FILE;
5358         goto end;
5359     }

5361     if (format == KMF_FORMAT_ASN1) {
5362         xcrl = d2i_X509_CRL_bio(in, NULL);
5363     } else if (format == KMF_FORMAT_PEM) {
5364         xcrl = PEM_read_bio_X509_CRL(in, NULL, NULL, NULL);
5365     }

5367     if (xcrl == NULL) {
5368         SET_ERROR(kmfh, ERR_get_error());
5369         ret = KMF_ERR_BAD_CRLFILE;
5370         goto end;
5371     }
5372     (void) BIO_free(in);

5374     /* Read the Certificate file and load it into a X509 structure */
5375     ret = kmf_is_cert_file(handle, certfile, &format);
5376     if (ret != KMF_OK)
5377         goto end;

5379     in = BIO_new_file(certfile, "rb");
5380     if (in == NULL) {
5381         SET_ERROR(kmfh, ERR_get_error());
5382         ret = KMF_ERR_OPEN_FILE;
5383         goto end;
5384     }

```

```
5386     if (format == KMF_FORMAT_ASN1) {
5387         xcert = d2i_X509_bio(in, NULL);
5388     } else if (format == KMF_FORMAT_PEM) {
5389         xcert = PEM_read_bio_X509(in, NULL, NULL, NULL);
5390     }
5392     if (xcert == NULL) {
5393         SET_ERROR(kmfh, ERR_get_error());
5394         ret = KMF_ERR_BAD_CERTFILE;
5395         goto end;
5396     }
5398     /* Check if the certificate and the CRL have same issuer */
5399     if (X509_NAME_cmp(xcert->cert_info->issuer, xcrl->crl->issuer) != 0) {
5400         ret = KMF_ERR_ISSUER;
5401         goto end;
5402     }
5404     /* Check to see if the certificate serial number is revoked */
5405     revoke_stack = X509_CRL_get_REVOKED(xcrl);
5406     if (sk_X509_REVOKED_num(revoke_stack) <= 0) {
5407         /* No revoked certificates in the CRL file */
5408         SET_ERROR(kmfh, ERR_get_error());
5409         ret = KMF_ERR_EMPTY_CRL;
5410         goto end;
5411     }
5413     for (i = 0; i < sk_X509_REVOKED_num(revoke_stack); i++) {
5427         /* LINTED E_BAD_PTR_CAST_ALIGN */
5414         revoke = sk_X509_REVOKED_value(revoke_stack, i);
5415         if (ASN1_INTEGER_cmp(xcert->cert_info->serialNumber,
5416             revoke->serialNumber) == 0) {
5417             break;
5418         }
5419     }
5421     if (i < sk_X509_REVOKED_num(revoke_stack)) {
5422         ret = KMF_OK;
5423     } else {
5424         ret = KMF_ERR_NOT_REVOKED;
5425     }
5427 end:
5428     if (in != NULL)
5429         (void) BIO_free(in);
5430     if (xcrl != NULL)
5431         X509_CRL_free(xcrl);
5432     if (xcert != NULL)
5433         X509_free(xcert);
5435     return (ret);
5436 }
unchanged_portion_omitted
```

```

*****
2531 Thu Feb 13 22:45:23 2014
new/usr/src/lib/libpkg/Makefile.com
don't pass in lint libraries which no longer exist
*****
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 #
23 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 LIBRARY= libpkg.a
27 VERS= .1
28
29 #
30 # include library definitions
31 OBJECTS= \
32     canonize.o ckparam.o ckvolseq.o \
33     devtype.o dstream.o gpkglist.o \
34     gpkgmap.o isdir.o logerr.o \
35     mappath.o ncgrpw.o nhash.o \
36     pkgexecl.o pkgexecv.o pkgmount.o \
37     pkgtrans.o ppkgmap.o \
38     progerr.o putcfiler.o rrmkdir.o \
39     runcmd.o srchcfile.o tputcfent.o \
40     verify.o security.o pkgweb.o \
41     pkgerr.o keystore.o pl2lib.o \
42     vfpop.o fmkdir.o pkgstr.o \
43     handlelocalfs.o pkgserv.o
44
45 # include library definitions
46 include $(SRC)/lib/Makefile.lib
47
48 SRCDIR= ../common
49
50 POFILE = libpkg.po
51 MSGFILES = $(OBJECTS:%.o=../common/%.i)
52 CLEANFILES += $(MSGFILES)
53
54 # This library is NOT lint clean
55
56 # openssl forces us to ignore dubious pointer casts, thanks to its clever
57 # use of macros for stack management.
58 LINTFLAGS= -umx -errtags \
59     -erroff=E_BAD_PTR_CAST_ALIGN,E_BAD_PTR_CAST
60 $(LINTLIB):= SRCS = $(SRCDIR)/$(LINTSRC)

```

```

64 LIBS = $(DYNLIB) $(LINTLIB)
65
66 LDLIBS += -lc -lwanboot -lscf -ladm
67
68 # libcrypto and libssl have no lint library, and so can only be used when
69 # building
70 $(DYNLIB) := -lcrypto -lssl
71 LDLIBS += -lc -lssl -lwanboot -lcrypto -lscf -ladm
72
73 CFLAGS += $(CCVERBOSE)
74 CERRWARN += -_gcc=-Wno-unused-label
75 CERRWARN += -_gcc=-Wno-parentheses
76 CERRWARN += -_gcc=-Wno-uninitialized
77 CERRWARN += -_gcc=-Wno-clobbered
78 CERRWARN += -_gcc=-Wno-switch
79 CERRWARN += -_gcc=-Wno-unused-value
80 CPPFLAGS += -I$(SRCDIR) -D_FILE_OFFSET_BITS=64
81
82 .KEEP_STATE:
83
84 all: $(LIBS)
85
86 $(POFILE): $(MSGFILES)
87     $(BUILDPO.msgfiles)
88
89 _msg: $(MSGDOMAINPOFILE)
90
91 lint: lintcheck
92
93 # include library targets
94 include $(SRC)/lib/Makefile.targ
95 include $(SRC)/Makefile.msg.targ

```

new/usr/src/lib/libshare/Makefile.com

1

```
*****
1904 Thu Feb 13 22:45:24 2014
new/usr/src/lib/libshare/Makefile.com
don't pass in lint libraries which no longer exist
*****
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 (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
22 #
23 LIBRARY =      libshare.a
24 VERS =        .1
25 NFSLIB_DIR =  $(SRC)/cmd/fs.d/nfs/lib

27 LIBOBS =      libshare.o libsharecore.o scfutil.o libshare_zfs.o \
28 plugin.o parser.o issubdir.o
29 OTHOBS =      sharetab.o nfs_sec.o
30 OBJECTS =     $(LIBOBS) $(OTHOBS)

32 include ../../Makefile.lib
33 SRCDIR =      ../common

35 LIBSRCS =     $(LIBOBS:%.o=$(SRCDIR)/%.c)
36 # we don't want to lint the sharetab and nfs_sec files
37 lintcheck := SRCS = $(LIBSRCS)

39 LIBS =        $(DYNLIB) $(LINTLIB)
40 LDLIBS +=     -lc -lnsl -lscf -lzfs -luuid -lnvpair
40 LDLIBS +=     -lc -lnsl -lscf -lzfs -luuid -lxml2 -lnvpair
41 $(LINTLIB) := SRCS = $(SRCDIR)/$(LINTSRC)

43 # libxml2 has no lint library, so we can only use it when building
44 $(DYNLIB) := LDLIBS += -lxml2

46 #endif /* ! codereview */
47 #add nfs/lib directory as part of the include path
48 CFLAGS +=     $(CVERBOSE)
49 CERRWARN +=   -_gcc=-Wno-parentheses
50 CERRWARN +=   -_gcc=-Wno-uninitialized
51 CERRWARN +=   -_gcc=-Wno-switch
52 CPPFLAGS +=   -D_REENTRANT -I$(NFSLIB_DIR) \
53              -I$(ADJUNCT_PROTO)/usr/include/libxml2

55 .KEEP_STATE:

57 all: $(LIBS)

59 lint: lintcheck
```

new/usr/src/lib/libshare/Makefile.com

2

```
61 pics/%.o:    $(NFSLIB_DIR)/%.c
62              $(COMPILE.c) -o $@ $<
63              $(POST_PROCESS_O)

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

new/usr/src/lib/libshare/autofs/Makefile.com

1

```
*****
2055 Thu Feb 13 22:45:24 2014
new/usr/src/lib/libshare/autofs/Makefile.com
don't pass in lint libraries which no longer exist
*****
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) 2010, Oracle and/or its affiliates. All rights reserved.
23 #
24 #

26 LIBRARY =      libshare_autofs.a
27 VERS =        .1
28 AUTOFSSMFLIB_DIR = $(SRC)/cmd/fs.d/nfs/lib

30 LIBOBS =      libshare_autofs.o
31 OTHOBS =      smfcfg.o
32 OBJECTS =     $(LIBOBS) $(OTHOBS)

34 include ../../Makefile.lib

36 ROOTLIBDIR =  $(ROOT)/usr/lib/fs/autofs
37 ROOTLIBDIR64 = $(ROOT)/usr/lib/fs/autofs/$(MACH64)

39 LIBSRCS = $(LIBOBS:%.o=$(SRCDIR)/%.c)
40 # we don't want to lint the sources for OTHOBS since they are pre-existing file
41 # that are not lint free.
42 lintcheck := SRCS = $(LIBSRCS)

44 LIBS =        $(DYNLIB)
45 LDLIBS +=     -lshare -lscf -lumem -lc

47 # libxml2 has no lint library, so we can only use it when building
48 $(DYNLIB) := LDLIBS += -lxml2
45 LDLIBS +=     -lshare -lscf -lumem -lc -lxml2

50 #add nfs/lib directory as part of the include path
51 CFLAGS +=     $(CVERBOSE)
52 CERRWARN +=   -_gcc=-Wno-switch
53 CERRWARN +=   -_gcc=-Wno-unused-variable
54 CERRWARN +=   -_gcc=-Wno-uninitialized
55 CPPFLAGS +=   -D_REENTRANT -I$(AUTOFSSMFLIB_DIR) \
56               -I$(ADJUNCT_PROTO)/usr/include/libxml2 \
57               -I$(SRCDIR)../common

59 .KEEP_STATE:
```

new/usr/src/lib/libshare/autofs/Makefile.com

2

```
61 all: $(LIBS)

63 install: $(ROOTLIBDIR) $(ROOTLIBDIR64) all

65 lint: lintcheck

67 pics/%.o:      $(AUTOFSSMFLIB_DIR)/%.c
68               $(COMPILE.c) -o $@ $<
69               $(POST_PROCESS_O)

71 $(ROOTLIBDIR):
72     $(INS.dir)
73
74 $(ROOTLIBDIR64):
75     $(INS.dir)

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

new/usr/src/lib/libshare/nfs/Makefile.com

1

```
*****
2011 Thu Feb 13 22:45:25 2014
new/usr/src/lib/libshare/nfs/Makefile.com
don't pass in lint libraries which no longer exist
*****
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 #
23 # Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
24 #

26 LIBRARY =      libshare_nfs.a
27 VERS =        .1
28 NFSLIB_DIR =   = $(SRC)/cmd/fs.d/nfs/lib

30 LIBOBS =      libshare_nfs.o
31 OTHOBS =      nfs_sec.o nfslog_config.o nfslogtab.o smfcfg.o
32 OBJECTS =     $(LIBOBS) $(OTHOBS)

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

36 ROOTLIBDIR =  $(ROOT)/usr/lib/fs/nfs
37 ROOTLIBDIR64 = $(ROOT)/usr/lib/fs/nfs/$(MACH64)

39 LIBSRCS = $(LIBOBS:%.o=$(SRCDIR)/%.c)
40 # we don't want to lint the sources for OTHOBS since they are pre-existing file
41 # that are not lint free.
42 lintcheck := SRCS = $(LIBSRCS)

44 LIBS =        $(DYNLIB)
45 LDLIBS +=     -lshare -lnsl -lscf -lumem -lc

47 # libxml2 has no lint library, so we can only use it when building
48 $(DYNLIB) := LDLIBS += -lxml2
45 LDLIBS +=     -lshare -lnsl -lscf -lumem -lc -lxml2

50 #add nfs/lib directory as part of the include path
51 CFLAGS +=     $(CVERBOSE)
52 CERRWARN +=   -_gcc=-Wno-parentheses
53 CERRWARN +=   -_gcc=-Wno-switch
54 CERRWARN +=   -_gcc=-Wno-unused-variable
55 CERRWARN +=   -_gcc=-Wno-uninitialized
56 CPPFLAGS +=   -D_REENTRANT -I$(NFSLIB_DIR) \
57              -I$(ADJUNCT_PROTO)/usr/include/libxml2 -I$(SRCDIR)/../../common

59 .KEEP_STATE:
```

new/usr/src/lib/libshare/nfs/Makefile.com

2

```
61 all: $(LIBS)

63 install: all

65 lint: lintcheck

67 pics/%.o: $(NFSLIB_DIR)/%.c
68             $(COMPILE.c) -o $@ $<
69             $(POST_PROCESS_0)

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

new/usr/src/lib/libshare/smbfs/Makefile.com

1

```
*****
1762 Thu Feb 13 22:45:25 2014
new/usr/src/lib/libshare/smbfs/Makefile.com
don't pass in lint libraries which no longer exist
*****
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 #

26 LIBRARY =      libshare_smbfs.a
27 VERS =        .1

29 LIBOBJS =      libshare_smbfs.o
30 SMBBASE_OBJ =  smbfs_scfutil.o
31 OBJECTS =      $(LIBOBJS) $(SMBBASE_OBJ)

33 include ../../Makefile.lib

35 ROOTLIBDIR =   $(ROOT)/usr/lib/fs/smbfs
36 ROOTLIBDIR64 = $(ROOT)/usr/lib/fs/smbfs/$(MACH64)

38 LIBSRCS = $(LIBOBJS:%.o=$(SRCDIR)/%.c)

40 LIBS =         $(DYNLIB)
41 LDLIBS +=      -lshare -lscf -lumem -luuid -lc -lsmbfs

43 # libxml2 has no lint library, so we can only use it when building
44 $(DYNLIB) := LDLIBS += -lxml2
41 LDLIBS +=      -lshare -lscf -lumem -luuid -lc -lxml2 -lsmbfs

46 CFLAGS +=     $(CVERBOSE)
47 CERRWARN +=   -_gcc=-Wno-switch
48 CERRWARN +=   -_gcc=-Wno-uninitialized
49 CPPFLAGS +=   -D_REENTRANT -I$(ADJUNCT_PROTO)/usr/include/libxml2 \
50              -I$(SRCDIR)/../common -I$(SRC)/lib/libshare -I$(SRC)/uts/common

52 .KEEP_STATE:

54 all: $(LIBS)

56 install: $(ROOTLIBDIR) $(ROOTLIBDIR64) all

58 lint: lintcheck

60 $(ROOTLIBDIR):
```

new/usr/src/lib/libshare/smbfs/Makefile.com

2

```
61      $(INS.dir)
63 $(ROOTLIBDIR64):
64      $(INS.dir)

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



\*\*\*\*\*

2761 Thu Feb 13 22:45:25 2014

new/usr/src/lib/libwanboot/Makefile.com

don't pass in lint libraries which no longer exist

\*\*\*\*\*

```

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 2009 Sun Microsystems, Inc. All rights reserved.
22 # Use is subject to license terms.
23 #
24 # Copyright (c) 2012 by Delphix. All rights reserved.
25 #

27 LIBRARY = libwanboot.a
28 VERS = .1

30 # List of locally located modules.
31 LOC_DIR = ../common
32 LOC_OBJS = socket_inet.o bootinfo_aux.o
33 LOC_SRCS = $(LOC_OBJS:%.o=$(LOC_DIR)/%.c)

35 # List of common wanboot objects.
36 COM_DIR = ../../../common/net/wanboot
37 COM_OBJS = auxutil.o \
38 boot_http.o \
39 bootconf.o \
40 bootconf_errmsg.o \
41 bootinfo.o \
42 bootlog.o \
43 http_errorstr.o \
44 pl2access.o \
45 pl2auxpars.o \
46 pl2auxutl.o \
47 pl2err.o \
48 pl2misc.o \
49 parseURL.o
50 COM_SRCS = $(COM_OBJS:%.o=$(COM_DIR)/%.c)

52 # List of common DHCP modules.
53 DHCP_DIR = $(SRC)/common/net/dhcp
54 DHCP_OBJS = dhcpinfo.o
55 DHCP_SRCS = $(DHCP_OBJS:%.o=$(DHCP_DIR)/%.c)

57 OBJECTS = $(LOC_OBJS) $(COM_OBJS) $(DHCP_OBJS)

59 include ../../Makefile.lib

61 LIBS += $(LINTLIB)

```

```

62 LDLIBS += -lnvpair -lresolv -lnsl -lsocket -ldevinfo -ldhcputil \
63 -linetutil -lc

65 # libcrypto and libssl have no lint library, so we can only use it when
66 # building
67 $(DYNLIB) := LDLIBS += -lcrypto -lssl

63 -linetutil -lc -lcrypto -lssl
69 CPPFLAGS = -I$(SRC)/common/net/wanboot/crypt $(CPPFLAGS.master)
70 CERRWARN += -_gcc=-Wno-switch
71 CERRWARN += -_gcc=-Wno-parentheses
72 CERRWARN += -_gcc=-Wno-unused-value
73 CERRWARN += -_gcc=-Wno-uninitialized

75 # Must override SRCS from Makefile.lib since sources have
76 # multiple source directories.
77 SRCS = $(LOC_SRCS) $(COM_SRCS) $(DHCP_SRCS)

79 # Must define location of lint library source.
80 SRCDIR = $(LOC_DIR)
81 $(LINTLIB) := SRCS = $(SRCDIR)/$(LINTSRC)

83 # OpenSSL requires us to turn this off
84 LINTFLAGS += -erroff=E_BAD_PTR_CAST_ALIGN
85 LINTFLAGS64 += -erroff=E_BAD_PTR_CAST_ALIGN

87 CFLAGS += $(CCVERBOSE)
88 CPPFLAGS += -I$(LOC_DIR) -I$(COM_DIR) -I$(DHCP_DIR)

90 .KEEP_STATE:

92 all: $(LIBS)

94 lint: lintcheck

96 pics/%.o: $(COM_DIR)/%.c
97 $(COMPILE.c) -o $@ $<
98 $(POST_PROCESS_O)

100 pics/%.o: $(DHCP_DIR)/%.c
101 $(COMPILE.c) -o $@ $<
102 $(POST_PROCESS_O)

104 include ../../Makefile.targ

```

new/usr/src/lib/pkcs11/pkcs11\_tpm/Makefile.com

1

\*\*\*\*\*

2467 Thu Feb 13 22:45:26 2014

new/usr/src/lib/pkcs11/pkcs11\_tpm/Makefile.com

don't pass in lint libraries which no longer exist

\*\*\*\*\*

```
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 2009 Sun Microsystems, Inc. All rights reserved.
22 # Use is subject to license terms.
23 #
24 LIBRARY =      pkcs11_tpm.a
25 VERS =        .1
```

```
27 OBJECTS= api_interface.o \
28   apiutil.o \
29   asnl.o \
30   cert.o \
31   data_obj.o \
32   decr_mgr.o \
33   dig_mgr.o \
34   encr_mgr.o \
35   globals.o \
36   hwf_obj.o \
37   key.o \
38   key_mgr.o \
39   loadsave.o \
40   log.o \
41   mech_md5.o \
42   mech_rsa.o \
43   mech_sha.o \
44   new_host.o \
45   obj_mgr.o \
46   object.o \
47   sess_mgr.o \
48   sign_mgr.o \
49   template.o \
50   tpm_specific.o \
51   utility.o \
52   verify_mgr.o
```

55 include \$(SRC)/lib/Makefile.lib

57 SRCDIR= ../common

59 SRCS= \$(OBJECTS:%.o=\$(SRCDIR)/%.c)

61 # set signing mode

new/usr/src/lib/pkcs11/pkcs11\_tpm/Makefile.com

2

62 POST\_PROCESS\_SO += ; \$(ELFSIGN\_CRYPT0)

64 ROOTLIBDIR=\$(ROOT)/usr/lib/security

65 ROOTLIBDIR64=\$(ROOT)/usr/lib/security/\$(MACH64)

67 LIBS=\$(DYNLIB) \$(DYNLIB64)

69 TSSROOT=\$(ADJUNCT\_PROTO)

70 TSPILIBDIR=\$(TSSROOT)/usr/lib

71 TSPINCDIR=\$(TSSROOT)/usr/include

72 TSSLIB=-L\$(TSPILIBDIR)

73 TSSLIB64=-L\$(TSPILIBDIR)/\$(MACH64)

74 TSSINC=-I\$(TSPINCDIR)

76 LDLIBS += \$(TSSLIB) -L\$(ADJUNCT\_PROTO)/lib -lc -luuid -lmd -ltspi -lcrypto

78 # libtspi and libcrypto have no lint library, so we can only use it when

79 # building

80 \$(LIBS) := LDLIBS += -ltspi -lcrypto

82 #endif /\* ! codereview \*/

83 CPPFLAGS += -xCC -D\_POSIX\_PTHREAD\_SEMANTICS \$(TSSINC)

84 CPPFLAGS64 += \$(CPPFLAGS)

85 C99MODE= \$(C99\_ENABLE)

87 CERRWARN += -\_gcc=-Wno-parentheses

88 CERRWARN += -\_gcc=-Wno-unused-label

89 CERRWARN += -\_gcc=-Wno-uninitialized

91 LINTSRC= \$(OBJECTS:%.o=\$(SRCDIR)/%.c)

93 \$(LINTLIB) := SRCS = \$(SRCDIR)/\$(LINTSRC)

94 LINTSRC= \$(SRCS)

96 CLOBBERFILES += C.ln

98 .KEEP\_STATE:

100 all: \$(LIBS)

101

102 lint: \$\$\$(LINTSRC)

103 \$(LINT.c) \$(LINTCHECKFLAGS) \$(LINTSRC) \$(LDLIBS)

105 pics/%.o: \$(SRCDIR)/%.c

106 \$(COMPILE.c) -o \$@ \$<

107 \$(POST\_PROCESS\_O)

109 include \$(SRC)/lib/Makefile.targ

```

*****
12578 Thu Feb 13 22:45:26 2014
new/usr/src/man/man1/Makefile
remove machid manual pages, since machid was removed
*****
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet
9 # at http://www.illumos.org/license/CDDL.
10 #
11 #
12 #
13 # Copyright 2011, Richard Lowe
14 # Copyright 2012 OmniTI Computer Consulting, Inc. All rights reserved.
15 # Copyright 2013 Nexenta Systems, Inc. All rights reserved.
16 #
17 #
18 include      $(SRC)/Makefile.master
19 #
20 MANSECT=     1
21 #
22 MANFILES=    acctcom.1  \
23              adb.1      \
24              addbib.1   \
25              alias.1    \
26              allocate.1 \
27              amt.1      \
28              appcert.1  \
29              apptrace.1 \
30              apropos.1  \
31              ar.1       \
32              arch.1     \
33              asa.1      \
34              at.1       \
35              atq.1      \
36              atrm.1     \
37              audioconvert.1 \
38              audiocvt.1 \
39              audioplay.1 \
40              audiorecord.1 \
41              audiotest.1 \
42              auths.1    \
43              awk.1      \
44              banner.1   \
45              basename.1 \
46              bc.1       \
47              bdiff.1    \
48              bfs.1      \
49              break.1    \
50              builtin.1  \
51              cal.1      \
52              calendar.1 \
53              cancel.1   \
54              cat.1      \
55              cd.1       \
56              cdrw.1     \
57              checknr.1  \
58              chgrp.1    \
59              chkey.1    \
60              chmod.1    \
61              chown.1

```

```

62          ckdate.1      \
63          ckgid.1       \
64          ckint.1       \
65          ckitem.1      \
66          ckkeywd.1     \
67          ckpath.1      \
68          ckrange.1     \
69          ckstr.1       \
70          cksum.1       \
71          cktime.1     \
72          ckuid.1      \
73          ckyorn.1     \
74          clear.1       \
75          cmp.1         \
76          col.1        \
77          comm.1        \
78          command.1    \
79          compress.1   \
80          cp.1          \
81          cpio.1        \
82          cputrack.1   \
83          crle.1       \
84          crontab.1    \
85          crypt.1      \
86          csh.1        \
87          csplit.1     \
88          ctags.1      \
89          ctrun.1      \
90          ctstat.1     \
91          ctwatch.1    \
92          cut.1        \
93          date.1       \
94          dc.1         \
95          deallocate.1 \
96          deroff.1     \
97          dhcpinfo.1   \
98          diff.1       \
99          diff3.1      \
100         diffmk.1     \
101         digest.1     \
102         dircmp.1     \
103         dis.1        \
104         disown.1     \
105         dispgid.1    \
106         dispuid.1    \
107         dos2unix.1   \
108         download.1   \
109         dpost.1      \
110         du.1         \
111         dump.1       \
112         dumpcs.1     \
113         echo.1       \
114         ed.1         \
115         egrep.1      \
116         eject.1      \
117         elfdump.1    \
118         elfedit.1    \
119         elfsign.1    \
120         elfwrap.1    \
121         enable.1     \
122         encrypt.1    \
123         enhance.1    \
124         env.1        \
125         eqn.1        \
126         exec.1       \
127         exit.1

```

```

128 expand.1
129 expr.1
130 exstr.1
131 factor.1
132 fdformat.1
133 fgrep.1
134 file.1
135 filebench.1
136 filesync.1
137 find.1
138 finger.1
139 fmt.1
140 fmtmsg.1
141 fold.1
142 ftp.1
143 ftpcount.1
144 ftpwho.1
145 gcore.1
146 gencat.1
147 genmsg.1
148 getconf.1
149 getfacl.1
150 getlabel.1
151 getopt.1
152 getoptcv.1
153 getopts.1
154 gettext.1
155 gettxt.1
156 getzonepath.1
157 glob.1
158 gprof.1
159 grep.1
160 groups.1
161 hash.1
162 head.1
163 history.1
164 hostid.1
165 hostname.1
166 iconv.1
167 indxbib.1
168 Intro.1
169 ipcrm.1
170 ipcs.1
171 isainfo.1
172 isalist.1
173 jobs.1
174 join.1
175 kbd.1
176 kdestroy.1
177 keylogin.1
178 keylogout.1
179 kill.1
180 kinit.1
181 klist.1
182 kmdb.1
183 kmfcfg.1
184 kpasswd.1
185 krb5-config.1
186 ksh93.1
187 ktutil.1
188 lari.1
189 last.1
190 lastcomm.1
191 ld.1
192 ldap.1
193 ldapdelete.1

```

```

194 ldaplist.1
195 ldapmodify.1
196 ldapmodrdn.1
197 ldapsearch.1
198 ldd.1
199 ld.so.1.1
200 let.1
201 lex.1
202 lgrpinfo.1
203 limit.1
204 line.1
205 list_devices.1
206 listusers.1
207 ln.1
208 loadkeys.1
209 locale.1
210 localedef.1
211 logger.1
212 login.1
213 logname.1
214 logout.1
215 look.1
216 lookbib.1
217 lorder.1
218 lp.1
219 lpstat.1
220 ls.1
221 m4.1
222 mac.1
223 mach.1
224 madv.so.1.1
225 mail.1
226 mailcompat.1
227 mailq.1
228 mailstats.1
229 mailx.1
230 makekey.1
231 man.1
232 mconnect.1
233 mcs.1
234 mdb.1
235 msg.1
236 mkdir.1
237 mkmsgs.1
238 mktemp.1
239 moe.1
240 more.1
241 mpss.so.1.1
242 msgcc.1
243 msgcpp.1
244 msgcv.1
245 msgfmt.1
246 msggen.1
247 msgget.1
248 mt.1
249 mv.1
250 nawk.1
251 nc.1
252 nca.1
253 ncab2clf.1
254 ncakmod.1
255 newform.1
256 newgrp.1
257 news.1
258 newtask.1
259 nice.1

```

```

260      nl.1
261      nm.1
262      nohup.1
263      nroff.1
264      od.1
265      on.1
266      optisa.1
267      pack.1
268      pagesize.1
269      pargs.1
270      passwd.1
271      paste.1
272      pathchk.1
273      pax.1
274      pfexec.1
275      pg.1
276      pgrep.1
277      pkginfo.1
278      pkgmk.1
279      pkgparam.1
280      pkgproto.1
281      pkgtrans.1
282      pktool.1
283      plabel.1
284      plgrp.1
285      plimit.1
286      pmadvise.1
287      pmap.1
288      postio.1
289      postprint.1
290      postreverse.1
291      ppgsz.1
292      ppriv.1
293      pr.1
294      praliases.1
295      prctl.1
296      preap.1
297      prex.1
298      print.1
299      printf.1
300      priocntl.1
301      proc.1
302      prof.1
303      profiles.1
304      projects.1
305      ps.1
306      ptree.1
307      pvs.1
308      pwd.1
309      ranlib.1
310      rcapstat.1
311      rcp.1
312      rdist.1
313      read.1
314      readonly.1
315      refer.1
316      regcmp.1
317      renice.1
318      rev.1
319      rlogin.1
320      rm.1
321      rmformat.1
322      rmmount.1
323      roffbib.1
324      roles.1
325      rpcgen.1

```

```

326      rsh.1
327      runat.1
328      rup.1
329      ruptime.1
330      rusers.1
331      rwho.1
332      sar.1
333      scp.1
334      script.1
335      sdiff.1
336      sed.1
337      set.1
338      setfacl.1
339      setlabel.1
340      setpgrp.1
341      sftp.1
342      shcomp.1
343      shell_builtins.1
344      shift.1
345      size.1
346      sleep.1
347      smbutil.1
348      soelim.1
349      sort.1
350      sortbib.1
351      sotruss.1
352      spell.1
353      split.1
354      srchtxt.1
355      ssh.1
356      ssh-add.1
357      ssh-agent.1
358      ssh-http-proxy-connect.1
359      ssh-keygen.1
360      ssh-keyscan.1
361      ssh-socks5-proxy-connect.1
362      strchg.1
363      strings.1
364      strip.1
365      stty.1
366      sum.1
367      suspend.1
368      svcprop.1
369      svcs.1
370      symorder.1
371      sys-suspend.1
372      tabs.1
373      tail.1
374      talk.1
375      tar.1
376      tbl.1
377      tcopy.1
378      tee.1
379      telnet.1
380      test.1
381      tftp.1
382      time.1
383      times.1
384      timex.1
385      tip.1
386      tnfdump.1
387      tnfxtract.1
388      touch.1
389      tput.1
390      tr.1
391      trap.1

```

```

392 troff.1 //
393 true.1 //
394 truss.1 //
395 tsort.1 //
396 tty.1 //
397 type.1 //
398 typeset.1 //
399 ul.1 //
400 umask.1 //
401 uname.1 //
402 unifdef.1 //
403 uniq.1 //
404 units.1 //
405 unix2dos.1 //
406 uptime.1 //
407 vacation.1 //
408 vgrind.1 //
409 volcheck.1 //
410 volrmmount.1 //
411 w.1 //
412 wait.1 //
413 wc.1 //
414 whatis.1 //
415 which.1 //
416 who.1 //
417 whocalls.1 //
418 whois.1 //
419 write.1 //
420 xargs.1 //
421 xgettext.1 //
422 xstr.1 //
423 yacc.1 //
424 yes.1 //
425 ypcat.1 //
426 ypmatch.1 //
427 yppasswd.1 //
428 ypwhich.1 //
429 zlogin.1 //
430 zonenname.1 //

432 MANLINKS= batch.1 //
433 bg.1 //
434 case.1 //
435 chdir.1 //
436 checkeq.1 //
437 continue.1 //
438 decrypt.1 //
439 dirname.1 //
440 dirs.1 //
441 disable.1 //
442 dumpkeys.1 //
443 edit.1 //
444 errange.1 //
445 errdate.1 //
446 errgid.1 //
447 errint.1 //
448 erritem.1 //
449 errpath.1 //
450 errstr.1 //
451 errtime.1 //
452 erruid.1 //
453 erryorn.1 //
454 eval.1 //
455 export.1 //
456 false.1 //
457 fc.1 //

```

```

458 fg.1 //
459 for.1 //
460 foreach.1 //
461 function.1 //
462 goto.1 //
463 hashcheck.1 //
464 hashmake.1 //
465 hashstat.1 //
466 helpdate.1 //
467 helpgid.1 //
468 helpint.1 //
469 helpitem.1 //
470 helppath.1 //
471 helprange.1 //
472 helpstr.1 //
473 helptime.1 //
474 helpuid.1 //
475 helpyorn.1 //
476 hist.1 //
477 if.1 //
478 intro.1 //
479 jsh.1 //
480 ksh.1 //
481 ldapadd.1 //
482 neqn.1 //
483 notify.1 //
484 onintr.1 //
485 page.1 //
486 pcat.1 //
487 pcred.1 //
488 pfcsh.1 //
489 pfiles.1 //
490 pfksh.1 //
491 pflags.1 //
492 pfsh.1 //
493 pkill.1 //
494 pldd.1 //
495 popd.1 //
496 prun.1 //
497 psig.1 //
498 pstack.1 //
499 pstop.1 //
500 ptime.1 //
501 pushd.1 //
502 pwait.1 //
503 pwdx.1 //
504 red.1 //
505 rehash.1 //
506 remote_shell.1 //
507 remsh.1 //
508 repeat.1 //
509 return.1 //
510 rksh.1 //
511 rksh93.1 //
512 rmail.1 //
513 rmdir.1 //
514 rmumount.1 //
515 select.1 //
516 setenv.1 //
517 settime.1 //
518 sh.1 //
519 snca.1 //
520 source.1 //
521 spellin.1 //
522 stop.1 //
523 strconf.1 //

```

```

524         switch.1          \|
525         ulimit.1         \|
526         unalias.1       \|
527         uncompress.1    \|
528         unexpand.1     \|
529         unhash.1       \|
530         unlimit.1      \|
531         unpack.1       \|
532         unset.1        \|
533         unsetenv.1     \|
534         until.1        \|
535         valdate.1      \|
536         valgid.1       \|
537         valint.1       \|
538         valpath.1      \|
539         valrange.1     \|
540         valstr.1       \|
541         valtime.1      \|
542         valuid.1       \|
543         valyorn.1     \|
544         vax.1          \|
545         vedit.1        \|
546         whence.1      \|
547         while.1       \|
548         zcat.1        \|

550 intro.1      := LINKSRC = Intro.1

552 unalias.1    := LINKSRC = alias.1

554 batch.1     := LINKSRC = at.1

556 dirname.1   := LINKSRC = basename.1

558 continue.1  := LINKSRC = break.1

560 chdir.1     := LINKSRC = cd.1
561 dirs.1      := LINKSRC = cd.1
562 popd.1      := LINKSRC = cd.1
563 pushd.1     := LINKSRC = cd.1

565 errdate.1   := LINKSRC = ckdate.1
566 helpdate.1 := LINKSRC = ckdate.1
567 valdate.1   := LINKSRC = ckdate.1

569 errgid.1    := LINKSRC = ckgid.1
570 helpgid.1  := LINKSRC = ckgid.1
571 valgid.1    := LINKSRC = ckgid.1

573 errint.1    := LINKSRC = ckint.1
574 helpint.1  := LINKSRC = ckint.1
575 valint.1    := LINKSRC = ckint.1

577 erritem.1   := LINKSRC = ckitem.1
578 helpitem.1 := LINKSRC = ckitem.1

580 errpath.1   := LINKSRC = ckpath.1
581 helppath.1 := LINKSRC = ckpath.1
582 valpath.1   := LINKSRC = ckpath.1

584 errrange.1  := LINKSRC = ckrange.1
585 helprange.1 := LINKSRC = ckrange.1
586 valrange.1  := LINKSRC = ckrange.1

588 errstr.1    := LINKSRC = ckstr.1
589 helpstr.1   := LINKSRC = ckstr.1

```

```

590 valstr.1    := LINKSRC = ckstr.1

592 errtime.1   := LINKSRC = cktime.1
593 helptime.1 := LINKSRC = cktime.1
594 valtime.1   := LINKSRC = cktime.1

596 erruid.1    := LINKSRC = ckuid.1
597 helpuid.1   := LINKSRC = ckuid.1
598 valuid.1    := LINKSRC = ckuid.1

600 erryorn.1   := LINKSRC = ckyorn.1
601 helpyorn.1 := LINKSRC = ckyorn.1
602 valyorn.1   := LINKSRC = ckyorn.1

604 uncompress.1 := LINKSRC = compress.1
605 zcat.1       := LINKSRC = compress.1

607 red.1       := LINKSRC = ed.1

609 disable.1   := LINKSRC = enable.1

611 decrypt.1   := LINKSRC = encrypt.1

613 checkeq.1   := LINKSRC = eqn.1
614 neqn.1      := LINKSRC = eqn.1

616 eval.1      := LINKSRC = exec.1
617 source.1    := LINKSRC = exec.1

619 goto.1      := LINKSRC = exit.1
620 return.1    := LINKSRC = exit.1

622 unexpand.1 := LINKSRC = expand.1

624 hashstat.1 := LINKSRC = hash.1
625 rehash.1   := LINKSRC = hash.1
626 unhash.1   := LINKSRC = hash.1

628 fc.1       := LINKSRC = history.1
629 hist.1     := LINKSRC = history.1

631 bg.1       := LINKSRC = jobs.1
632 fg.1       := LINKSRC = jobs.1
633 notify.1   := LINKSRC = jobs.1
634 stop.1     := LINKSRC = jobs.1

636 jsh.1      := LINKSRC = ksh93.1
637 ksh.1      := LINKSRC = ksh93.1
638 rksh.1     := LINKSRC = ksh93.1
639 rksh93.1   := LINKSRC = ksh93.1
640 sh.1       := LINKSRC = ksh93.1

642 ldapadd.1 := LINKSRC = ldapmodify.1

644 ulimit.1   := LINKSRC = limit.1
645 unlimit.1  := LINKSRC = limit.1

647 dumpkeys.1 := LINKSRC = loadkeys.1

649 i286.1     := LINKSRC = machid.1
650 i386.1     := LINKSRC = machid.1
651 i486.1     := LINKSRC = machid.1
652 i860.1     := LINKSRC = machid.1
653 iAPX286.1 := LINKSRC = machid.1
654 pdp11.1    := LINKSRC = machid.1
655 sparc.1    := LINKSRC = machid.1

```

```

656 sun.1          := LINKSRC = machid.1
657 u370.1         := LINKSRC = machid.1
658 u3b.1          := LINKSRC = machid.1
659 u3b15.1        := LINKSRC = machid.1
660 u3b2.1         := LINKSRC = machid.1
661 u3b5.1         := LINKSRC = machid.1
662 vax.1          := LINKSRC = machid.1

649 rmail.1       := LINKSRC = mail.1

651 page.1        := LINKSRC = more.1

653 snca.1        := LINKSRC = nca.1

655 pcat.1        := LINKSRC = pack.1
656 unpack.1      := LINKSRC = pack.1

658 pfcsh.1       := LINKSRC = pfexec.1
659 pfksh.1       := LINKSRC = pfexec.1
660 pfsh.1        := LINKSRC = pfexec.1

662 pkill.1       := LINKSRC = pgrep.1

664 pcred.1       := LINKSRC = proc.1
665 pfiles.1      := LINKSRC = proc.1
666 pflags.1      := LINKSRC = proc.1
667 pldd.1        := LINKSRC = proc.1
668 prun.1        := LINKSRC = proc.1
669 psig.1        := LINKSRC = proc.1
670 pstack.1      := LINKSRC = proc.1
671 pstop.1       := LINKSRC = proc.1
672 ptime.1       := LINKSRC = proc.1
673 pwait.1       := LINKSRC = proc.1
674 pwdx.1        := LINKSRC = proc.1

676 rmdir.1       := LINKSRC = rm.1

678 rmumount.1    := LINKSRC = rmmount.1

680 remote_shell.1 := LINKSRC = rsh.1
681 remsh.1       := LINKSRC = rsh.1

683 export.1      := LINKSRC = set.1
684 setenv.1      := LINKSRC = set.1
685 unset.1       := LINKSRC = set.1
686 unsetenv.1    := LINKSRC = set.1

688 case.1        := LINKSRC = shell_builtins.1
689 for.1          := LINKSRC = shell_builtins.1
690 foreach.1      := LINKSRC = shell_builtins.1
691 function.1     := LINKSRC = shell_builtins.1
692 if.1          := LINKSRC = shell_builtins.1
693 repeat.1       := LINKSRC = shell_builtins.1
694 select.1       := LINKSRC = shell_builtins.1
695 switch.1       := LINKSRC = shell_builtins.1
696 until.1        := LINKSRC = shell_builtins.1
697 while.1        := LINKSRC = shell_builtins.1

699 hashcheck.1   := LINKSRC = spell.1
700 hashmake.1    := LINKSRC = spell.1
701 spellin.1     := LINKSRC = spell.1

703 strconf.1     := LINKSRC = strchg.1

705 settime.1     := LINKSRC = touch.1

```

```

707 onintr.1      := LINKSRC = trap.1

709 false.1       := LINKSRC = true.1

711 whence.1      := LINKSRC = typeset.1

713 # Links to usr/has/man

715 edit.1        := LINKSRC = ../../../has/man/man1has/edit.lhas

717 vedit.1       := LINKSRC = ../../../has/man/man1has/vi.lhas

719 .KEEP_STATE:

721 include        $(SRC)/man/Makefile.man

723 install:      $(ROOTMANFILES) $(ROOTMANLINKS)

```



new/usr/src/pkg/manifests/SUNWcs.man1.inc

1

\*\*\*\*\*

13156 Thu Feb 13 22:45:27 2014

new/usr/src/pkg/manifests/SUNWcs.man1.inc

remove machid manual pages, since machid was removed

\*\*\*\*\*

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet
9 # at http://www.illumos.org/license/CDDL.
10 #
11 #
12 #
13 # Copyright 2011, Richard Lowe
14 # Copyright 2012 OmniTI Computer Consulting, Inc. All rights reserved.
15 # Copyright 2012 Nexenta Systems, Inc. All rights reserved.
16 #
```

```
18 file path=usr/share/man/man1/Intro.1
19 file path=usr/share/man/man1/alias.1
20 file path=usr/share/man/man1/allocate.1
21 file path=usr/share/man/man1/amt.1
22 file path=usr/share/man/man1/arch.1
23 file path=usr/share/man/man1/at.1
24 file path=usr/share/man/man1/atq.1
25 file path=usr/share/man/man1/atrm.1
26 file path=usr/share/man/man1/auths.1
27 file path=usr/share/man/man1/basename.1
28 file path=usr/share/man/man1/break.1
29 file path=usr/share/man/man1/builtin.1
30 file path=usr/share/man/man1/cat.1
31 file path=usr/share/man/man1/cd.1
32 file path=usr/share/man/man1/chgrp.1
33 file path=usr/share/man/man1/chmod.1
34 file path=usr/share/man/man1/chown.1
35 file path=usr/share/man/man1/ckdate.1
36 file path=usr/share/man/man1/ckgid.1
37 file path=usr/share/man/man1/ckint.1
38 file path=usr/share/man/man1/ckitem.1
39 file path=usr/share/man/man1/ckkeywd.1
40 file path=usr/share/man/man1/ckpath.1
41 file path=usr/share/man/man1/ckrange.1
42 file path=usr/share/man/man1/ckstr.1
43 file path=usr/share/man/man1/cksum.1
44 file path=usr/share/man/man1/cktime.1
45 file path=usr/share/man/man1/ckuid.1
46 file path=usr/share/man/man1/ckyorn.1
47 file path=usr/share/man/man1/clear.1
48 file path=usr/share/man/man1/cmp.1
49 file path=usr/share/man/man1/comm.1
50 file path=usr/share/man/man1/command.1
51 file path=usr/share/man/man1/cp.1
52 file path=usr/share/man/man1/cpio.1
53 file path=usr/share/man/man1/crle.1
54 file path=usr/share/man/man1/crontab.1
55 file path=usr/share/man/man1/crypt.1
56 file path=usr/share/man/man1/csh.1
57 file path=usr/share/man/man1/ctrunc.1
58 file path=usr/share/man/man1/ctstat.1
59 file path=usr/share/man/man1/ctwatch.1
60 file path=usr/share/man/man1/cut.1
61 file path=usr/share/man/man1/date.1
```

new/usr/src/pkg/manifests/SUNWcs.man1.inc

2

```
62 file path=usr/share/man/man1/deallocate.1
63 file path=usr/share/man/man1/dhccpinfo.1
64 file path=usr/share/man/man1/digest.1
65 file path=usr/share/man/man1/disown.1
66 file path=usr/share/man/man1/dispuid.1
67 file path=usr/share/man/man1/dispuid.1
68 file path=usr/share/man/man1/du.1
69 file path=usr/share/man/man1/dumpcs.1
70 file path=usr/share/man/man1/echo.1
71 file path=usr/share/man/man1/ed.1
72 file path=usr/share/man/man1/egrep.1
73 file path=usr/share/man/man1/eject.1
74 file path=usr/share/man/man1/encrypt.1
75 file path=usr/share/man/man1/env.1
76 file path=usr/share/man/man1/exec.1
77 file path=usr/share/man/man1/exit.1
78 file path=usr/share/man/man1/expr.1
79 file path=usr/share/man/man1/fdformat.1
80 file path=usr/share/man/man1/fgrep.1
81 file path=usr/share/man/man1/file.1
82 file path=usr/share/man/man1/find.1
83 file path=usr/share/man/man1/fmt.1
84 file path=usr/share/man/man1/fmtmsg.1
85 file path=usr/share/man/man1/fold.1
86 file path=usr/share/man/man1/getconf.1
87 file path=usr/share/man/man1/getfacl.1
88 file path=usr/share/man/man1/getopt.1
89 file path=usr/share/man/man1/getoptcv.1
90 file path=usr/share/man/man1/getopts.1
91 file path=usr/share/man/man1/gettext.1
92 file path=usr/share/man/man1/glob.1
93 file path=usr/share/man/man1/grep.1
94 file path=usr/share/man/man1/groups.1
95 file path=usr/share/man/man1/hash.1
96 file path=usr/share/man/man1/head.1
97 file path=usr/share/man/man1/history.1
98 file path=usr/share/man/man1/hostid.1
99 file path=usr/share/man/man1/hostname.1
100 file path=usr/share/man/man1/iconv.1
101 file path=usr/share/man/man1/isainfo.1
102 file path=usr/share/man/man1/isalist.1
103 file path=usr/share/man/man1/jobs.1
104 file path=usr/share/man/man1/join.1
105 file path=usr/share/man/man1/kbd.1
106 file path=usr/share/man/man1/keylogin.1
107 file path=usr/share/man/man1/keylogout.1
108 file path=usr/share/man/man1/kill.1
109 file path=usr/share/man/man1/kmfcfg.1
110 file path=usr/share/man/man1/ksh93.1
111 file path=usr/share/man/man1/ld.so.1.1
112 file path=usr/share/man/man1/let.1
113 file path=usr/share/man/man1/limit.1
114 file path=usr/share/man/man1/line.1
115 file path=usr/share/man/man1/list_devices.1
116 file path=usr/share/man/man1/listusers.1
117 file path=usr/share/man/man1/ln.1
118 file path=usr/share/man/man1/loadkeys.1
119 file path=usr/share/man/man1/localedef.1
120 file path=usr/share/man/man1/logger.1
121 file path=usr/share/man/man1/login.1
122 file path=usr/share/man/man1/logname.1
123 file path=usr/share/man/man1/logout.1
124 file path=usr/share/man/man1/ls.1
125 file path=usr/share/man/man1/m4.1
126 file path=usr/share/man/man1/mac.1
127 file path=usr/share/man/man1/mach.1
```

```

128 file path=usr/share/man/man1/mail.1
129 file path=usr/share/man/man1/mailx.1
130 file path=usr/share/man/man1/makekey.1
131 file path=usr/share/man/man1/mesg.1
132 file path=usr/share/man/man1/mkdir.1
133 file path=usr/share/man/man1/mktemp.1
134 file path=usr/share/man/man1/moe.1
135 file path=usr/share/man/man1/more.1
136 file path=usr/share/man/man1/mt.1
137 file path=usr/share/man/man1/mv.1
138 file path=usr/share/man/man1/newgrp.1
139 file path=usr/share/man/man1/newtask.1
140 file path=usr/share/man/man1/nice.1
141 file path=usr/share/man/man1/nohup.1
142 file path=usr/share/man/man1/optisa.1
143 file path=usr/share/man/man1/pagesize.1
144 file path=usr/share/man/man1/passwd.1
145 file path=usr/share/man/man1/paste.1
146 file path=usr/share/man/man1/pathchk.1
147 file path=usr/share/man/man1/pax.1
148 file path=usr/share/man/man1/pfexec.1
149 file path=usr/share/man/man1/pg.1
150 file path=usr/share/man/man1/pgrep.1
151 file path=usr/share/man/man1/pktool.1
152 file path=usr/share/man/man1/pr.1
153 file path=usr/share/man/man1/prctl.1
154 file path=usr/share/man/man1/print.1
155 file path=usr/share/man/man1/priocntl.1
156 file path=usr/share/man/man1/proc.1
157 file path=usr/share/man/man1/profiles.1
158 file path=usr/share/man/man1/projects.1
159 file path=usr/share/man/man1/ps.1
160 file path=usr/share/man/man1/pwd.1
161 file path=usr/share/man/man1/read.1
162 file path=usr/share/man/man1/readonly.1
163 file path=usr/share/man/man1/renice.1
164 file path=usr/share/man/man1/rev.1
165 file path=usr/share/man/man1/rm.1
166 file path=usr/share/man/man1/roles.1
167 file path=usr/share/man/man1/runat.1
168 file path=usr/share/man/man1/script.1
169 file path=usr/share/man/man1/sed.1
170 file path=usr/share/man/man1/set.1
171 file path=usr/share/man/man1/setfacl.1
172 file path=usr/share/man/man1/setpgrp.1
173 file path=usr/share/man/man1/shcomp.1
174 file path=usr/share/man/man1/shell_builtins.1
175 file path=usr/share/man/man1/shift.1
176 file path=usr/share/man/man1/sleep.1
177 file path=usr/share/man/man1/strchg.1
178 file path=usr/share/man/man1/stty.1
179 file path=usr/share/man/man1/sum.1
180 file path=usr/share/man/man1/suspend.1
181 file path=usr/share/man/man1/svcprop.1
182 file path=usr/share/man/man1/svcs.1
183 file path=usr/share/man/man1/tabs.1
184 file path=usr/share/man/man1/tail.1
185 file path=usr/share/man/man1/tar.1
186 file path=usr/share/man/man1/tee.1
187 file path=usr/share/man/man1/test.1
188 file path=usr/share/man/man1/time.1
189 file path=usr/share/man/man1/times.1
190 file path=usr/share/man/man1/tip.1
191 file path=usr/share/man/man1/touch.1
192 file path=usr/share/man/man1/tput.1
193 file path=usr/share/man/man1/tr.1

```

```

194 file path=usr/share/man/man1/trap.1
195 file path=usr/share/man/man1/true.1
196 file path=usr/share/man/man1/tty.1
197 file path=usr/share/man/man1/type.1
198 file path=usr/share/man/man1/typeset.1
199 file path=usr/share/man/man1/umask.1
200 file path=usr/share/man/man1/uname.1
201 file path=usr/share/man/man1/uniq.1
202 file path=usr/share/man/man1/uptime.1
203 file path=usr/share/man/man1/w.1
204 file path=usr/share/man/man1/wait.1
205 file path=usr/share/man/man1/wc.1
206 file path=usr/share/man/man1/which.1
207 file path=usr/share/man/man1/who.1
208 file path=usr/share/man/man1/write.1
209 file path=usr/share/man/man1/xargs.1
210 file path=usr/share/man/man1/xstr.1
211 file path=usr/share/man/man1/zonename.1
212 link path=usr/share/man/man1/batch.1 target=at.1
213 link path=usr/share/man/man1/bg.1 target=jobs.1
214 link path=usr/share/man/man1/case.1 target=shell_builtins.1
215 link path=usr/share/man/man1/chdir.1 target=cd.1
216 link path=usr/share/man/man1/continue.1 target=break.1
217 link path=usr/share/man/man1/decrypt.1 target=encrypt.1
218 link path=usr/share/man/man1/dirname.1 target=basename.1
219 link path=usr/share/man/man1/dirs.1 target=cd.1
220 link path=usr/share/man/man1/dumpkeys.1 target=loadkeys.1
221 link path=usr/share/man/man1/edit.1 target=../../../../has/man/man1has/edit.1has
222 link path=usr/share/man/man1/errange.1 target=ckrange.1
223 link path=usr/share/man/man1/errdate.1 target=ckdate.1
224 link path=usr/share/man/man1/errgid.1 target=ckgid.1
225 link path=usr/share/man/man1/errint.1 target=ckint.1
226 link path=usr/share/man/man1/erritem.1 target=ckitem.1
227 link path=usr/share/man/man1/errpath.1 target=ckpath.1
228 link path=usr/share/man/man1/errstr.1 target=ckstr.1
229 link path=usr/share/man/man1/errtime.1 target=cktime.1
230 link path=usr/share/man/man1/erruid.1 target=ckuid.1
231 link path=usr/share/man/man1/erryorn.1 target=ckyorn.1
232 link path=usr/share/man/man1/eval.1 target=exec.1
233 link path=usr/share/man/man1/export.1 target=set.1
234 link path=usr/share/man/man1/false.1 target=true.1
235 link path=usr/share/man/man1/fc.1 target=history.1
236 link path=usr/share/man/man1/fg.1 target=jobs.1
237 link path=usr/share/man/man1/for.1 target=shell_builtins.1
238 link path=usr/share/man/man1/foreach.1 target=shell_builtins.1
239 link path=usr/share/man/man1/function.1 target=shell_builtins.1
240 link path=usr/share/man/man1/goto.1 target=exit.1
241 link path=usr/share/man/man1/hashstat.1 target=hash.1
242 link path=usr/share/man/man1/helpdate.1 target=ckdate.1
243 link path=usr/share/man/man1/helpgid.1 target=ckgid.1
244 link path=usr/share/man/man1/helpint.1 target=ckint.1
245 link path=usr/share/man/man1/helpitem.1 target=ckitem.1
246 link path=usr/share/man/man1/helppath.1 target=ckpath.1
247 link path=usr/share/man/man1/helprange.1 target=ckrange.1
248 link path=usr/share/man/man1/helpstr.1 target=ckstr.1
249 link path=usr/share/man/man1/helptime.1 target=cktime.1
250 link path=usr/share/man/man1/helpuid.1 target=ckuid.1
251 link path=usr/share/man/man1/helpyorn.1 target=ckyorn.1
252 link path=usr/share/man/man1/hist.1 target=history.1
253 link path=usr/share/man/man1/i286.1 target=machid.1
254 link path=usr/share/man/man1/i386.1 target=machid.1
255 link path=usr/share/man/man1/i486.1 target=machid.1
256 link path=usr/share/man/man1/i860.1 target=machid.1
257 link path=usr/share/man/man1/iAPX286.1 target=machid.1
253 link path=usr/share/man/man1/if.1 target=shell_builtins.1
254 link path=usr/share/man/man1/intro.1 target=Intro.1

```

```
255 link path=usr/share/man/man1/jsh.1 target=ksh93.1
256 link path=usr/share/man/man1/ksh.1 target=ksh93.1
257 link path=usr/share/man/man1/notify.1 target=jobs.1
258 link path=usr/share/man/man1/onintr.1 target=trap.1
259 link path=usr/share/man/man1/page.1 target=more.1
260 link path=usr/share/man/man1/pcred.1 target=proc.1
266 link path=usr/share/man/man1/pdp11.1 target=machid.1
261 link path=usr/share/man/man1/pfcsh.1 target=pfexec.1
262 link path=usr/share/man/man1/pfiles.1 target=proc.1
263 link path=usr/share/man/man1/pfsh.1 target=pfexec.1
264 link path=usr/share/man/man1/pflags.1 target=proc.1
265 link path=usr/share/man/man1/pfsh.1 target=pfexec.1
266 link path=usr/share/man/man1/pkill.1 target=pgrep.1
267 link path=usr/share/man/man1/pldd.1 target=proc.1
268 link path=usr/share/man/man1/popd.1 target=cd.1
269 link path=usr/share/man/man1/prun.1 target=proc.1
270 link path=usr/share/man/man1/psig.1 target=proc.1
271 link path=usr/share/man/man1/pstack.1 target=proc.1
272 link path=usr/share/man/man1/pstop.1 target=proc.1
273 link path=usr/share/man/man1/ptime.1 target=proc.1
274 link path=usr/share/man/man1/pushd.1 target=cd.1
275 link path=usr/share/man/man1/pwait.1 target=proc.1
276 link path=usr/share/man/man1/pwdx.1 target=proc.1
277 link path=usr/share/man/man1/red.1 target=ed.1
278 link path=usr/share/man/man1/rehash.1 target=hash.1
279 link path=usr/share/man/man1/repeat.1 target=shell_builtins.1
280 link path=usr/share/man/man1/return.1 target=exit.1
281 link path=usr/share/man/man1/rksh.1 target=ksh93.1
282 link path=usr/share/man/man1/rksh93.1 target=ksh93.1
283 link path=usr/share/man/man1/rmail.1 target=mail.1
284 link path=usr/share/man/man1/rmdir.1 target=rm.1
285 link path=usr/share/man/man1/select.1 target=shell_builtins.1
286 link path=usr/share/man/man1/setenv.1 target=set.1
287 link path=usr/share/man/man1/settime.1 target=touch.1
288 link path=usr/share/man/man1/sh.1 target=ksh93.1
289 link path=usr/share/man/man1/source.1 target=exec.1
296 link path=usr/share/man/man1/sparc.1 target=machid.1
290 link path=usr/share/man/man1/stop.1 target=jobs.1
291 link path=usr/share/man/man1/strconf.1 target=strchg.1
299 link path=usr/share/man/man1/sun.1 target=machid.1
292 link path=usr/share/man/man1/switch.1 target=shell_builtins.1
301 link path=usr/share/man/man1/u370.1 target=machid.1
302 link path=usr/share/man/man1/u3b.1 target=machid.1
303 link path=usr/share/man/man1/u3b15.1 target=machid.1
304 link path=usr/share/man/man1/u3b2.1 target=machid.1
305 link path=usr/share/man/man1/u3b5.1 target=machid.1
293 link path=usr/share/man/man1/ulimit.1 target=limit.1
294 link path=usr/share/man/man1/unalias.1 target=alias.1
295 link path=usr/share/man/man1/unhash.1 target=hash.1
296 link path=usr/share/man/man1/unlimit.1 target=limit.1
297 link path=usr/share/man/man1/unset.1 target=set.1
298 link path=usr/share/man/man1/unsetenv.1 target=set.1
299 link path=usr/share/man/man1/until.1 target=shell_builtins.1
300 link path=usr/share/man/man1/valdate.1 target=ckdate.1
301 link path=usr/share/man/man1/valgid.1 target=ckgid.1
302 link path=usr/share/man/man1/valint.1 target=ckint.1
303 link path=usr/share/man/man1/valpath.1 target=ckpath.1
304 link path=usr/share/man/man1/valrange.1 target=ckrange.1
305 link path=usr/share/man/man1/valstr.1 target=ckstr.1
306 link path=usr/share/man/man1/vaftime.1 target=cktime.1
307 link path=usr/share/man/man1/valuid.1 target=ckuid.1
308 link path=usr/share/man/man1/valyorn.1 target=ckyorn.1
322 link path=usr/share/man/man1/vax.1 target=machid.1
309 link path=usr/share/man/man1/vedit.1 target=.././has/man/man1has/vi.1has
310 link path=usr/share/man/man1/whence.1 target=typeset.1
311 link path=usr/share/man/man1/while.1 target=shell_builtins.1
```

```

*****
26469 Thu Feb 13 22:45:27 2014
new/usr/src/uts/common/os/grow.c
os/grow: define 'p' under the same ifdef as it's consumed
*****
_____unchanged_portion_omitted_____

492 /*
493  * Find address for user to map.
494  * If MAP_FIXED is not specified, we can pick any address we want, but we will
495  * first try the value in *addrp if it is non-NULL. Thus this is implementing
496  * a way to try and get a preferred address.
497  */
498 int
499 choose_addr(struct as *as, caddr_t *addrp, size_t len, offset_t off,
500             int vacalign, uint_t flags)
501 {
502 #if defined(__amd64)
503 #endif /* ! codereview */
504     proc_t *p = curproc;
505 #endif
506 #endif /* ! codereview */
507     caddr_t basep = (caddr_t)(uintptr_t)((uintptr_t)*addrp & PAGEMASK);
508     size_t lenp;

510     ASSERT(AS_ISCLAIMGAP(as)); /* searches should be serialized */

512     /*
513      * If we have been provided a hint, we should still expand the lenp
514      * to be the rest of the address space. This will allow us to
515      * treat the hint as a strong desire to be "nearby" the provided
516      * address. If we can't satisfy the hint, as_gap() will walk forward.
517      */
518     if (flags & _MAP_LOW32)
519         lenp = (caddr_t)USERLIMIT32 - basep;
520 #if defined(__amd64)
521     else if (p->p_model == DATAMODEL_NATIVE)
522         lenp = p->p_usrstack - basep -
523             ((p->p_stk_ctl + PAGEOFFSET) & PAGEMASK);
524 #endif
525     else
526         lenp = as->a_userlimit - basep;

528     if (flags & MAP_FIXED) {
529         (void) as_unmap(as, *addrp, len);
530         return (0);
531     } else if (basep != NULL && ((flags & MAP_ALIGN) == 0) &&
532             !as_gap(as, len, &basep, &lenp, 0, *addrp)) {
533         /* User supplied address was available */
534         *addrp = basep;
535     } else {
536         /*
537          * No user supplied address or the address supplied was not
538          * available.
539          */
540         map_addr(addrp, len, off, vacalign, flags);
541     }
542     if (*addrp == NULL)
543         return (ENOMEM);
544     return (0);
545 }

548 /*
549  * Used for MAP_ANON - fast way to get anonymous pages
550  */

```

```

551 static int
552 zmap(struct as *as, caddr_t *addrp, size_t len, uint_t uprot, int flags,
553       offset_t pos)
554 {
555     struct segvn_crargs vn_a;
556     int error;

558     if (((PROT_ALL & uprot) != uprot))
559         return (EACCES);

561     if ((flags & MAP_FIXED) != 0) {
562         caddr_t userlimit;

564         /*
565          * Use the user address. First verify that
566          * the address to be used is page aligned.
567          * Then make some simple bounds checks.
568          */
569         if (((uintptr_t)*addrp & PAGEOFFSET) != 0)
570             return (EINVAL);

572         userlimit = flags & _MAP_LOW32 ?
573             (caddr_t)USERLIMIT32 : as->a_userlimit;
574         switch (valid_usr_range(*addrp, len, uprot, as, userlimit)) {
575             case RANGE_OKAY:
576                 break;
577             case RANGE_BADPROT:
578                 return (ENOTSUP);
579             case RANGE_BADADDR:
580                 return (ENOMEM);
581             default:
582                 return (ENOMEM);
583         }
584     }
585     /*
586      * No need to worry about vac alignment for anonymous
587      * pages since this is a "clone" object that doesn't
588      * yet exist.
589      */
590     error = choose_addr(as, addrp, len, pos, ADDR_NOVACALIGN, flags);
591     if (error != 0) {
592         return (error);
593     }

594     /*
595      * Use the seg_vn segment driver; passing in the NULL amp
596      * gives the desired "cloning" effect.
597      */
598     vn_a.vp = NULL;
599     vn_a.offset = 0;
600     vn_a.type = flags & MAP_TYPE;
601     vn_a.prot = uprot;
602     vn_a.maxprot = PROT_ALL;
603     vn_a.flags = flags & ~MAP_TYPE;
604     vn_a.cred = CRED();
605     vn_a.amp = NULL;
606     vn_a.szc = 0;
607     vn_a.lgrp_mem_policy_flags = 0;

609     return (as_map(as, *addrp, len, segvn_create, &vn_a));
610 }

612 static int
613 smmap_common(caddr_t *addrp, size_t len,
614             int prot, int flags, struct file *fp, offset_t pos)
615 {
616     struct vnode *vp;

```

```

617 struct as *as = curproc->p_as;
618 uint_t uprot, maxprot, type;
619 int error;
620 int in_crit = 0;

622 if ((flags & ~(MAP_SHARED | MAP_PRIVATE | MAP_FIXED | _MAP_NEW |
623   _MAP_LOW32 | MAP_NORESERVE | MAP_ANON | MAP_ALIGN |
624   MAP_TEXT | MAP_INITDATA)) != 0) {
625     /* | MAP_RENAME */ /* not implemented, let user know */
626     return (EINVAL);
627 }

629 if ((flags & MAP_TEXT) && !(prot & PROT_EXEC)) {
630     return (EINVAL);
631 }

633 if ((flags & (MAP_TEXT | MAP_INITDATA)) == (MAP_TEXT | MAP_INITDATA)) {
634     return (EINVAL);
635 }

637 #if defined(__sparc)
638 /*
639  * See if this is an "old mmap call". If so, remember this
640  * fact and convert the flags value given to mmap to indicate
641  * the specified address in the system call must be used.
642  * _MAP_NEW is turned set by all new uses of mmap.
643  */
644 if ((flags & _MAP_NEW) == 0)
645     flags |= MAP_FIXED;
646 #endif
647 flags &= ~_MAP_NEW;

649 type = flags & MAP_TYPE;
650 if (type != MAP_PRIVATE && type != MAP_SHARED)
651     return (EINVAL);

654 if (flags & MAP_ALIGN) {
656     if (flags & MAP_FIXED)
657         return (EINVAL);

659     /* alignment needs to be a power of 2 >= page size */
660     if (((uintptr_t)*addrp < PAGE_SIZE && (uintptr_t)*addrp != 0) ||
661         !ISP2((uintptr_t)*addrp))
662         return (EINVAL);
663 }
664 /*
665  * Check for bad lengths and file position.
666  * We let the VOP_MAP routine check for negative lengths
667  * since on some vnode types this might be appropriate.
668  */
669 if (len == 0 || (pos & (u_offset_t)PAGEOFFSET) != 0)
670     return (EINVAL);

672 maxprot = PROT_ALL; /* start out allowing all accesses */
673 uprot = prot | PROT_USER;

675 if (fp == NULL) {
676     ASSERT(flags & MAP_ANON);
677     /* discard lwpchan mappings, like munmap() */
678     if ((flags & MAP_FIXED) && curproc->p_lcp != NULL)
679         lwpchan_delete_mapping(curproc, *addrp, *addrp + len);
680     as_rangelock(as);
681     error = zmap(as, addrp, len, uprot, flags, pos);
682     as_rangeunlock(as);

```

```

683     /*
684     * Tell machine specific code that lwp has mapped shared memory
685     */
686     if (error == 0 && (flags & MAP_SHARED)) {
687         /* EMPTY */
688         LWP_MMODEL_SHARED_AS(*addrp, len);
689     }
690     return (error);
691 } else if ((flags & MAP_ANON) != 0)
692     return (EINVAL);

694 vp = fp->f_vnode;

696 /* Can't execute code from "noexec" mounted filesystem. */
697 if ((vp->v_vfsp->vfs_flag & VFS_NOEXEC) != 0)
698     maxprot &= ~PROT_EXEC;

700 /*
701  * These checks were added as part of large files.
702  *
703  * Return ENXIO if the initial position is negative; return EOVERFLOW
704  * if (offset + len) would overflow the maximum allowed offset for the
705  * type of file descriptor being used.
706  */
707 if (vp->v_type == VREG) {
708     if (pos < 0)
709         return (ENXIO);
710     if ((offset_t)len > (OFFSET_MAX(fp) - pos))
711         return (EOVERFLOW);
712 }

714 if (type == MAP_SHARED && (fp->f_flag & FWRITE) == 0) {
715     /* no write access allowed */
716     maxprot &= ~PROT_WRITE;
717 }

719 /*
720  * XXX - Do we also adjust maxprot based on protections
721  * of the vnode? E.g. if no execute permission is given
722  * on the vnode for the current user, maxprot probably
723  * should disallow PROT_EXEC also? This is different
724  * from the write access as this would be a per vnode
725  * test as opposed to a per fd test for writability.
726  */

728 /*
729  * Verify that the specified protections are not greater than
730  * the maximum allowable protections. Also test to make sure
731  * that the file descriptor does allow for read access since
732  * "write only" mappings are hard to do since normally we do
733  * the read from the file before the page can be written.
734  */
735 if (((maxprot & uprot) != uprot) || (fp->f_flag & FREAD) == 0)
736     return (EACCES);

738 /*
739  * If the user specified an address, do some simple checks here
740  */
741 if ((flags & MAP_FIXED) != 0) {
742     caddr_t userlimit;

744     /*
745     * Use the user address. First verify that
746     * the address to be used is page aligned.
747     * Then make some simple bounds checks.
748     */

```

```

749     if (((uintptr_t)*addrp & PAGEOFFSET) != 0)
750         return (EINVAL);

752     userlimit = flags & _MAP_LOW32 ?
753         (caddr_t)USERLIMIT32 : as->a_userlimit;
754     switch (valid_usr_range(*addrp, len, uprot, as, userlimit)) {
755     case RANGE_OKAY:
756         break;
757     case RANGE_BADPROT:
758         return (ENOTSUP);
759     case RANGE_BADADDR:
760     default:
761         return (ENOMEM);
762     }
763 }

765 if ((prot & (PROT_READ | PROT_WRITE | PROT_EXEC)) &&
766     nbl_need_check(vp)) {
767     int svmand;
768     nbl_op_t nop;

770     nbl_start_crit(vp, RW_READER);
771     in_crit = 1;
772     error = nbl_svmand(vp, fp->f_cred, &svmand);
773     if (error != 0)
774         goto done;
775     if ((prot & PROT_WRITE) && (type == MAP_SHARED)) {
776         if (prot & (PROT_READ | PROT_EXEC)) {
777             nop = NBL_READWRITE;
778         } else {
779             nop = NBL_WRITE;
780         }
781     } else {
782         nop = NBL_READ;
783     }
784     if (nbl_conflict(vp, nop, 0, LONG_MAX, svmand, NULL)) {
785         error = EACCES;
786         goto done;
787     }
788 }

790 /* discard lwpchan mappings, like munmap() */
791 if ((flags & MAP_FIXED) && curproc->p_lcp != NULL)
792     lwpchan_delete_mapping(curproc, *addrp, *addrp + len);

794 /*
795  * Ok, now let the vnode map routine do its thing to set things up.
796  */
797 error = VOP_MAP(vp, pos, as,
798     addrp, len, uprot, maxprot, flags, fp->f_cred, NULL);

800 if (error == 0) {
801     /*
802      * Tell machine specific code that lwp has mapped shared memory
803      */
804     if (flags & MAP_SHARED) {
805         /* EMPTY */
806         LWP_MMODEL_SHARED_AS(*addrp, len);
807     }
808     if (vp->v_type == VREG &&
809         (flags & (MAP_TEXT | MAP_INITDATA)) != 0) {
810         /*
811          * Mark this as an executable vnode
812          */
813         mutex_enter(&vp->v_lock);
814         vp->v_flag |= VVMEEXEC;

```

```

815         mutex_exit(&vp->v_lock);
816     }
817 }

819 done:
820     if (in_crit)
821         nbl_end_crit(vp);
822     return (error);
823 }

825 #ifdef _LP64
826 /*
827  * LP64 mmap(2) system call: 64-bit offset, 64-bit address.
828  *
829  * The "large file" mmap routine mmap64(2) is also mapped to this routine
830  * by the 64-bit version of libc.
831  *
832  * Eventually, this should be the only version, and have smmap_common()
833  * folded back into it again. Some day.
834  */
835 caddr_t
836 smmap64(caddr_t addr, size_t len, int prot, int flags, int fd, off_t pos)
837 {
838     struct file *fp;
839     int error;

841     if (fd == -1 && (flags & MAP_ANON) != 0)
842         error = smmap_common(&addr, len, prot, flags,
843             NULL, (offset_t)pos);
844     else if ((fp = getf(fd)) != NULL) {
845         error = smmap_common(&addr, len, prot, flags,
846             fp, (offset_t)pos);
847         releasef(fd);
848     } else
849         error = EBADF;

851     return (error ? (caddr_t)(uintptr_t)set_errno(error) : addr);
852 }
853 #endif /* _LP64 */

855 #if defined(_SYSCALL32_IMPL) || defined(_ILP32)

857 /*
858  * ILP32 mmap(2) system call: 32-bit offset, 32-bit address.
859  */
860 caddr_t
861 smmap32(caddr32_t addr, size32_t len, int prot, int flags, int fd, off32_t pos)
862 {
863     struct file *fp;
864     int error;
865     caddr_t a = (caddr_t)(uintptr_t)addr;

867     if (flags & _MAP_LOW32)
868         error = EINVAL;
869     else if (fd == -1 && (flags & MAP_ANON) != 0)
870         error = smmap_common(&a, (size_t)len, prot,
871             flags | _MAP_LOW32, NULL, (offset_t)pos);
872     else if ((fp = getf(fd)) != NULL) {
873         error = smmap_common(&a, (size_t)len, prot,
874             flags | _MAP_LOW32, fp, (offset_t)pos);
875         releasef(fd);
876     } else
877         error = EBADF;

879     ASSERT(error != 0 || (uintptr_t)(a + len) < (uintptr_t)UINT32_MAX);

```

```

881     return (error ? (caddr_t)(uintptr_t)set_errno(error) : a);
882 }

884 /*
885  * ILP32 mmap64(2) system call: 64-bit offset, 32-bit address.
886  *
887  * Now things really get ugly because we can't use the C-style
888  * calling convention for more than 6 args, and 64-bit parameter
889  * passing on 32-bit systems is less than clean.
890  */

892 struct mmaplf32a {
893     caddr_t addr;
894     size_t len;
895 #ifdef LP64
896     /*
897      * 32-bit contents, 64-bit cells
898      */
899     uint64_t prot;
900     uint64_t flags;
901     uint64_t fd;
902     uint64_t offhi;
903     uint64_t offlo;
904 #else
905     /*
906      * 32-bit contents, 32-bit cells
907      */
908     uint32_t prot;
909     uint32_t flags;
910     uint32_t fd;
911     uint32_t offhi;
912     uint32_t offlo;
913 #endif
914 };

916 int
917 smmaplf32(struct mmaplf32a *uap, rval_t *rvp)
918 {
919     struct file *fp;
920     int error;
921     caddr_t a = uap->addr;
922     int flags = (int)uap->flags;
923     int fd = (int)uap->fd;
924 #ifdef BIG_ENDIAN
925     offset_t off = ((u_offset_t)uap->offhi << 32) | (u_offset_t)uap->offlo;
926 #else
927     offset_t off = ((u_offset_t)uap->offlo << 32) | (u_offset_t)uap->offhi;
928 #endif

930     if (flags & _MAP_LOW32)
931         error = EINVAL;
932     else if (fd == -1 && (flags & MAP_ANON) != 0)
933         error = smmap_common(&a, uap->len, (int)uap->prot,
934             flags | _MAP_LOW32, NULL, off);
935     else if ((fp = getf(fd)) != NULL) {
936         error = smmap_common(&a, uap->len, (int)uap->prot,
937             flags | _MAP_LOW32, fp, off);
938         releasef(fd);
939     } else
940         error = EBADF;

942     if (error == 0)
943         rvp->r_val1 = (uintptr_t)a;
944     return (error);
945 }

```

```

947 #endif /* _SYSCALL32_IMPL || _ILP32 */

949 int
950 munmap(caddr_t addr, size_t len)
951 {
952     struct proc *p = curproc;
953     struct as *as = p->p_as;

955     if (((uintptr_t)addr & PAGEOFFSET) != 0 || len == 0)
956         return (set_errno(EINVAL));

958     if (valid_usr_range(addr, len, 0, as, as->a_userlimit) != RANGE_OKAY)
959         return (set_errno(EINVAL));

961     /*
962      * Discard lwpchan mappings.
963      */
964     if (p->p_lcp != NULL)
965         lwpchan_delete_mapping(p, addr, addr + len);
966     if (as_unmap(as, addr, len) != 0)
967         return (set_errno(EINVAL));

969     return (0);
970 }

972 int
973 mprotect(caddr_t addr, size_t len, int prot)
974 {
975     struct as *as = curproc->p_as;
976     uint_t uprot = prot | PROT_USER;
977     int error;

979     if (((uintptr_t)addr & PAGEOFFSET) != 0 || len == 0)
980         return (set_errno(EINVAL));

982     switch (valid_usr_range(addr, len, prot, as, as->a_userlimit)) {
983     case RANGE_OKAY:
984         break;
985     case RANGE_BADPROT:
986         return (set_errno(ENOTSUP));
987     case RANGE_BADADDR:
988     default:
989         return (set_errno(ENOMEM));
990     }

992     error = as_setprot(as, addr, len, uprot);
993     if (error)
994         return (set_errno(error));
995     return (0);
996 }

998 #define MC_CACHE      128
999 #define MC_QUANTUM    (MC_CACHE * PAGESIZE) /* internal result buffer */
/* addresses covered in loop */

1001 int
1002 mincore(caddr_t addr, size_t len, char *vecp)
1003 {
1004     struct as *as = curproc->p_as;
1005     caddr_t ea; /* end address of loop */
1006     size_t rl; /* inner result length */
1007     char vec[MC_CACHE]; /* local vector cache */
1008     int error;
1009     model_t model;
1010     long llen;

1012     model = get_udatamodel();

```

```
1013      /*
1014      * Validate form of address parameters.
1015      */
1016      if (model == DATAMODEL_NATIVE) {
1017          llen = (long)len;
1018      } else {
1019          llen = (int32_t)(size32_t)len;
1020      }
1021      if (((uintptr_t)addr & PAGEOFFSET) != 0 || llen <= 0)
1022          return (set_errno(EINVAL));
1024
1025      if (valid_usr_range(addr, len, 0, as, as->a_userlimit) != RANGE_OKAY)
1026          return (set_errno(ENOMEM));
1027
1028      /*
1029      * Loop over subranges of interval [addr : addr + len), recovering
1030      * results internally and then copying them out to caller. Subrange
1031      * is based on the size of MC_CACHE, defined above.
1032      */
1033      for (ea = addr + len; addr < ea; addr += MC_QUANTUM) {
1034          error = as_incore(as, addr,
1035              (size_t)MIN(MC_QUANTUM, ea - addr), vec, &rl);
1036          if (rl != 0) {
1037              rl = (rl + PAGESIZE - 1) / PAGESIZE;
1038              if (copyout(vec, vecp, rl) != 0)
1039                  return (set_errno(EFAULT));
1040              vecp += rl;
1041          }
1042          if (error != 0)
1043              return (set_errno(ENOMEM));
1044      }
1045      return (0);
1046 }
```