

new/usr/src/Makefile.lint

1

```
*****
8706 Sun May 26 18:22:28 2013
new/usr/src/Makefile.lint
XXX Remove awk(1)
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
22 # Copyright (c) 2003, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright (c) 2012 by Delphix. All rights reserved.
25 # include global definitions
26 include Makefile.master
28 #
29 # As pieces are made lint-clean, add them here so the nightly build
30 # can be used to keep them that way.
31 #
32 COMMON_SUBDIRS = \
33     cmd/acctadm \
34     cmd/asa \
35     cmd/amt \
36     cmd/audio/audiocctl \
37     cmd/audio/audiotest \
38     cmd/audit \
39     cmd/auditconfig \
40     cmd/auditd \
41     cmd/auditreduce \
42     cmd/auditstat \
43     cmd/auths \
44     cmd/autopush \
45     cmd/availdevs \
46     cmd/avs \
47     cmd/awk \
47     cmd/banner \
48     cmd/bart \
49     cmd/basename \
50     cmd/bdiff \
51     cmd/bfs \
52     cmd/busstat \
53     cmd/boot \
54     cmd/cal \
55     cmd/captoinfo \
56     cmd/cat \
57     cmd/cdrw \
58     cmd/cfgadm \
59     cmd/checkeq \
60     cmd/checknr \
```

new/usr/src/Makefile.lint

2

```
61     cmd/chgrp \
62     cmd/chmod \
63     cmd/chown \
64     cmd/chroot \
65     cmd/clinfo \
66     cmd/cmd-crypto \
67     cmd/cmd-inet/lib \
68     cmd/cmd-inet/lib/netcfgd \
69     cmd/cmd-inet/lib/nwamd \
70     cmd/cmd-inet/sbin \
71     cmd/cmd-inet/usr.bin \
72     cmd/cmd-inet/usr.lib/bridged \
73     cmd/cmd-inet/usr.lib/dsvclockd \
74     cmd/cmd-inet/usr.lib/ilbd \
75     cmd/cmd-inet/usr.lib/in.dhcpd \
76     cmd/cmd-inet/usr.lib/in.mpathd \
77     cmd/cmd-inet/usr.lib/in.ndpd \
78     cmd/cmd-inet/usr.lib/inetd \
79     cmd/cmd-inet/usr.lib/pppoe \
80     cmd/cmd-inet/usr.lib/slpd \
81     cmd/cmd-inet/usr.lib/vrrpd \
82     cmd/cmd-inet/usr.lib/wpad \
83     cmd/cmd-inet/usr.lib/wanboot \
84     cmd/cmd-inet/usr.sadm \
85     cmd/cmd-inet/usr.sbin \
86     cmd/cmd-inet/usr.sbin/ilbadm \
87     cmd/cmd-inet/usr.sbin/nwamadm \
88     cmd/cmd-inet/usr.sbin/nwamcfg \
89     cmd/col \
90     cmd/compress \
91     cmd/consadm \
92     cmd/coreadm \
93     cmd/cpc \
94     cmd/cpio \
95     cmd/crypt \
96     cmd/csplit \
97     cmd/ctrun \
98     cmd/ctstat \
99     cmd/ctwatch \
100    cmd/date \
101    cmd/dd \
102    cmd/deroff \
103    cmd/devctl \
104    cmd/devfsadm \
105    cmd/devinfo \
106    cmd/devmgmt \
107    cmd/devprop \
108    cmd/dfs.cmds \
109    cmd/diff3 \
110    cmd/dis \
111    cmd/dirname \
112    cmd/diskscan \
113    cmd/dispadm \
114    cmd/dladm \
115    cmd/dlmgmt \
116    cmd/dtrace \
117    cmd/du \
118    cmd/dumpadm \
119    cmd/dumpcs \
120    cmd/echo \
121    cmd/eject \
122    cmd/emul64ioctl \
123    cmd/env \
124    cmd/expand \
125    cmd/fcinfo \
126    cmd/fdetach \
```

new/usr/src/Makefile.lint

```

127 cmd/fdformat \
128 cmd/fdisk \
129 cmd/fgrep \
130 cmd/file \
131 cmd/filebench \
132 cmd/find \
133 cmd/fmthard \
134 cmd/fmtmsg \
135 cmd/fold \
136 cmd/fm \
137 cmd/format \
138 cmd/fs.d/fd \
139 cmd/fs.d/lofs/mount \
140 cmd/fs.d/mntfs \
141 cmd/fs.d/pcfs/mount \
142 cmd/fs.d/proc \
143 cmd/fs.d/tmpfs \
144 cmd/fs.d/udfs/mount \
145 cmd/fs.d/ufs/mount \
146 cmd/fs.d/ufs/fsirand \
147 cmd/fs.d/zfs/fstyp \
148 cmd/fwflash \
149 cmd/fuser \
150 cmd/gcore \
151 cmd/genmsg \
152 cmd/getconf \
153 cmd/getdevpolicy \
154 cmd/getfacl \
155 cmd/getopt \
156 cmd/gettext \
157 cmd/grep \
158 cmd/grep_xpg4 \
159 cmd/groups \
160 cmd/halt \
161 cmd/head \
162 cmd/hostid \
163 cmd/hostname \
164 cmd/hotplug \
165 cmd/hotplugd \
166 cmd/idmap \
167 cmd/init \
168 cmd/intrstat \
169 cmd/ipcrm \
170 cmd/ipcs \
171 cmd/isaexec \
172 cmd/isalist \
173 cmd/iscsiadm \
174 cmd/iscsid \
175 cmd/iscsitsvc \
176 cmd/isns \
177 cmd/itadm \
178 cmd/kbd \
179 cmd/killall \
180 cmd/ldap \
181 cmd/last \
182 cmd/lastcomm \
183 cmd/ldapcachemgr \
184 cmd/line \
185 cmd/link \
186 cmd/locator \
187 cmd/localedef \
188 cmd/lockstat \
189 cmd/lofiadm \
190 cmd/logadm \
191 cmd/logger \
192 cmd/login \

```

3

new/usr/src/Makefile.lint

```

193 cmd/logins \
194 cmd/ls \
195 cmd/luxadm \
196 cmd/lvm \
197 cmd/machid \
198 cmd/makekey \
199 cmd/mdb \
200 cmd/mesg \
201 cmd/mkdir \
202 cmd/mkfifo \
203 cmd/mkfile \
204 cmd/mkmsgs \
205 cmd/mknod \
206 cmd/mpathadm \
207 cmd/modload \
208 cmd/msgfmt \
209 cmd/msggid \
210 cmd/mt \
211 cmd/mv \
212 cmd/ndmpadm \
213 cmd/ndmpd \
214 cmd/ndmpstat \
215 cmd/newform \
216 cmd/newgrp \
217 cmd/newtask \
218 cmd/nice \
219 cmd/nl \
220 cmd/nohup \
221 cmd/nscd \
222 cmd/od \
223 cmd/pagesize \
224 cmd/passwd \
225 cmd/pathchk \
226 cmd/pbind \
227 cmd/pcidr \
228 cmd/pcitool \
229 cmd/pfexec \
230 cmd/pgrep \
231 cmd/picl/picld \
232 cmd/picl/prtpicl \
233 cmd/plockstat \
234 cmd/pools \
235 cmd/power \
236 cmd/powertop \
237 cmd/printf \
238 cmd/latencytop \
239 cmd/ppgsz \
240 cmd/praudit \
241 cmd/prctl \
242 cmd/priocntl \
243 cmd/profiles \
244 cmd/prstat \
245 cmd/prtconf \
246 cmd/prtdiag \
247 cmd/prvtoc \
248 cmd/ps \
249 cmd/psradm \
250 cmd/psrinfo \
251 cmd/psrset \
252 cmd/ptools \
253 cmd/pwck \
254 cmd/pwconv \
255 cmd/ramdiskadm \
256 cmd/raidctl \
257 cmd/rcap \
258 cmd/rcm_daemon \

```

4

new/usr/src/Makefile.lint

```

259 cmd/rctladm \
260 cmd/renice \
261 cmd/rm \
262 cmd/rmdir \
263 cmd/rmformat \
264 cmd/rmt \
265 cmd/roles \
266 cmd/rpcgen \
267 cmd/rpcsvc/rpc.bootparamd \
268 cmd/runat \
269 cmd/savecore \
270 cmd/sbdadm \
271 cmd/sdpadm \
272 cmd/sed \
273 cmd/setpgrp \
274 cmd/smbios \
275 cmd/sgs \
276 cmd/smsbrv \
277 cmd/smsrverd \
278 cmd/sort \
279 cmd/split \
280 cmd/srptadm \
281 cmd/srptsvc \
282 cmd/ssh \
283 cmd/stat \
284 cmd/stmfadm \
285 cmd/stmfsvc \
286 cmd/stmsboot \
287 cmd/streams/strcmd \
288 cmd/strings \
289 cmd/su \
290 cmd/sulogin \
291 cmd/svc \
292 cmd/swap \
293 cmd/sync \
294 cmd/syseventadm \
295 cmd/syseventd \
296 cmd/syslogd \
297 cmd/tabs \
298 cmd/tail \
299 cmd/th_tools \
300 cmd/tip \
301 cmd/touch \
302 cmd/tr \
303 cmd/truss \
304 cmd/tty \
305 cmd/tzreload \
306 cmd/uadmin \
307 cmd/ul \
308 cmd/userattr \
309 cmd/users \
310 cmd/utmp_update \
311 cmd/utmpd \
312 cmd/valtools \
313 cmd/vrrpadm \
314 cmd/vt \
315 cmd/wall \
316 cmd/who \
317 cmd/whodo \
318 cmd/wracct \
319 cmd/wusbadm \
320 cmd/xargs \
321 cmd/xstr \
322 cmd/yes \
323 cmd/yppasswd \
324 cmd/zdb \

```

5

new/usr/src/Makefile.lint

```

325 cmd/zdump \
326 cmd/zfs \
327 cmd/zhack \
328 cmd/zinject \
329 cmd/zlogin \
330 cmd/zoneadm \
331 cmd/zoneadmd \
332 cmd/zonectg \
333 cmd/zonename \
334 cmd/zpool \
335 cmd/zlook \
336 cmd/ztest \
337 lib/abi \
338 lib/auditd_plugins \
339 lib/libbe \
340 lib/pylibbe \
341 lib/brand/snl \
342 lib/brand/solaris10 \
343 lib/crypt_modules \
344 lib/extendedFILE \
345 lib/libadm \
346 lib/libadutils \
347 lib/libadt_jni \
348 lib/libaio \
349 lib/libavl \
350 lib/libbrand \
351 lib/libbsdmalloc \
352 lib/libbsm \
353 lib/libc \
354 lib/libc_db \
355 lib/libcfdgm \
356 lib/libcmdutils \
357 lib/libcommputil \
358 lib/libcontract \
359 lib/libcryptoutil \
360 lib/libctf \
361 lib/libdevice \
362 lib/libdevvid \
363 lib/libdevinfo \
364 lib/libdhcpage \
365 lib/libdhcpcdu \
366 lib/libdhcpsvc \
367 lib/libdhcputil \
368 lib/libdisasm \
369 lib/libdiskmgmt \
370 lib/libdladm \
371 lib/libdlpi \
372 lib/libdoor \
373 lib/libdscfg \
374 lib/libdtrace \
375 lib/libefi \
376 lib/libelfsign \
377 lib/libexacct \
378 lib/libfcoe \
379 lib/libgen \
380 lib/libgrubmgmt \
381 lib/libgss \
382 lib/libhotplug \
383 lib/libidmap \
384 lib/libilb \
385 lib/libinetsvc \
386 lib/libinetutil \
387 lib/libinstzones \
388 lib/libipadm \
389 lib/libipmi \
390 lib/libipmp \

```

6

new/usr/src/Makefile.lint

```

391 lib/libbipp \
392 lib/libipsecutil \
393 lib/libiscsit \
394 lib/libkfm \
395 lib/libkstat \
396 lib/liblgrp \
397 lib/liblm \
398 lib/libmalloc \
399 lib/libmapmalloc \
400 lib/libmapid \
401 lib/libmd \
402 lib/libmp \
403 lib/libmtmalloc \
404 lib/libndmp \
405 lib/libnsctl \
406 lib/libnsl \
407 lib/libnvpair \
408 lib/libnwam \
409 lib/libpam \
410 lib/libpctx \
411 lib/libpicl \
412 lib/libpicltree \
413 lib/libpkg \
414 lib/libpool \
415 lib/libproc \
416 lib/libpthread \
417 lib/libraidcfg \
418 lib/librcm \
419 lib/librdc \
420 lib/libreparse \
421 lib/librestart \
422 lib/librstp \
423 lib/librt \
424 lib/libscf \
425 lib/libsec \
426 lib/libsecdb \
427 lib/libsendfile \
428 lib/libsip \
429 lib/libshare \
430 lib/libsldap \
431 lib/libslp \
432 lib/libmbfs \
433 lib/libmbios \
434 lib/libsmmedia \
435 lib/libsrpt \
436 lib/libstmf \
437 lib/libsun_ima \
438 lib/libsysevent \
439 lib/libthread \
440 lib/libtsnet \
441 lib/libtsol \
442 lib/libumem \
443 lib/libunistat \
444 lib/libuuid \
445 lib/libuutil \
446 lib/libvrrpadm \
447 lib/libwanboot \
448 lib/libwanbootutil \
449 lib/libxnet \
450 lib/libzfs \
451 lib/libzfs_jni \
452 lib/libzonecfg \
453 lib/libzoneinfo \
454 lib/lvm \
455 lib/madv \
456 lib/mpss \

```

7

new/usr/src/Makefile.lint

```

457 lib/nametoaddr \
458 lib/ncad_addr \
459 lib/nsswitch \
460 lib/pam_modules \
461 lib/passwdutil \
462 lib/pkcs11 \
463 lib/print \
464 lib/raidcfg_plugins \
465 lib/scsi \
466 lib/smsrv \
467 lib/fm \
468 lib/udapl \
469 lib/watchmalloc \
470 psm \
471 test \
472 ucbscmd/basename \
473 ucbscmd/biff \
474 ucbscmd/echo \
475 ucbscmd/groups \
476 ucbscmd/mkstr \
477 ucbscmd/printenv \
478 ucbscmd/sum \
479 ucbscmd/test \
480 ucbscmd/users \
481 ucbscmd/whoami

483 i386_SUBDIRS= \
484 cmd/acpihp \
485 cmd/biosdev \
486 cmd/rtc \
487 cmd/ucodeadm \
488 lib/cfgadm_plugins/sata \
489 lib/cfgadm_plugins/sbd \
490 lib/libfdisk

492 sparc_SUBDIRS= \
493 cmd/datadm \
494 cmd/dcs \
495 cmd/drd \
496 cmd/fruadm \
497 cmd/ldmad \
498 cmd/prtdsc \
499 cmd/prtfu \
500 cmd/sckmd \
501 cmd/virtinfo \
502 cmd/vntsd \
503 lib/libds \
504 lib/libdsc \
505 lib/libpri \
506 lib/libpcc \
507 lib/libtsalarm \
508 lib/libv12n \
509 lib/storage \
510 stand

512 LINTSUBDIRS= $(COMMON_SUBDIRS) $(MACH)_SUBDIRS

514 .PARALLEL: $(LINTSUBDIRS)

516 lint: uts .WAIT subdirs

518 subdirs: $(LINTSUBDIRS)

520 uts $(LINTSUBDIRS): FRC
521 @cd $@; pwd; $(MAKE) lint

```

8

new/usr/src/Makefile.lint

9

523 FRC:

new/usr/src/Makefile.master

1

```
*****
35954 Sun May 26 18:22:31 2013
new/usr/src/Makefile.master
XXX Remove awk(1)
*****
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) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2012 by Delphix. All rights reserved.
25 #
26 #
27 #
28 # Makefile.master, global definitions for system source
29 #
30 ROOT=          /proto
31 #
32 #
33 # RELEASE_BUILD should be cleared for final release builds.
34 # NOT_RELEASE_BUILD is exactly what the name implies.
35 #
36 # INTERNAL_RELEASE_BUILD is a subset of RELEASE_BUILD. It mostly controls
37 # identification strings. Enabling RELEASE_BUILD automatically enables
38 # INTERNAL_RELEASE_BUILD.
39 #
40 # EXPORT_RELEASE_BUILD controls whether binaries are built in a form that
41 # can be released for export under a binary license. It is orthogonal to
42 # the other *RELEASE_BUILD settings. ("#" means do an export release
43 # build, "" means do a normal build.)
44 #
45 # CLOSED_BUILD controls whether we try to build files under
46 # usr/closed. (" means to build closed code, "#" means don't try to
47 # build it.) Skipping the closed code implies doing an export release
48 # build.
49 #
50 # STRIP_COMMENTS toggles comment section stripping. Generally the same setting
51 # as INTERNAL_RELEASE_BUILD.
52 #
53 # __GNUC toggles the building of ON components using gcc and related tools.
54 # Normally set to '#', set it to '' to do gcc build.
55 #
56 # The declaration POUND_SIGN is always '#'. This is needed to get around the
57 # make feature that '#' is always a comment delimiter, even when escaped or
58 # quoted. We use this macro expansion method to get POUND_SIGN rather than
59 # always breaking out a shell because the general case can cause a noticeable
60 # slowdown in build times when so many Makefiles include Makefile.master.
61 #
```

new/usr/src/Makefile.master

2

```
62 # While the majority of users are expected to override the setting below
63 # with an env file (via nightly or bldenv), if you aren't building that way
64 # (ie, you're using "ws" or some other bootstrapping method) then you need
65 # this definition in order to avoid the subshell invocation mentioned above.
66 #
67 #
68 PRE_POUND=          pre\#
69 POUND_SIGN=         $(PRE_POUND:pre\%=%)
70 #
71 NOT_RELEASE_BUILD=
72 INTERNAL_RELEASE_BUILD=      $(POUND_SIGN)
73 RELEASE_BUILD=          $(POUND_SIGN)
74 $(RELEASE_BUILD)NOT_RELEASE_BUILD=      $(POUND_SIGN)
75 $(RELEASE_BUILD)INTERNAL_RELEASE_BUILD=  $(POUND_SIGN)
76 PATCH_BUILD=           $(POUND_SIGN)
77 #
78 # If CLOSED_IS_PRESENT is not set, assume the closed tree is present.
79 CLOSED_BUILD_1= $(CLOSED_IS_PRESENT=yes)
80 CLOSED_BUILD=   $(CLOSED_BUILD_1:no=$(POUND_SIGN))
81 #
82 EXPORT_RELEASE_BUILD=      $(POUND_SIGN)
83 $(CLOSED_BUILD)EXPORT_RELEASE_BUILD=
84 #
85 # SPARC_BLD is '#' for an Intel build.
86 # INTEL_BLD is '#' for a Sparc build.
87 SPARC_BLD_1= $(MACH:i386=$(POUND_SIGN))
88 SPARC_BLD=   $(SPARC_BLD_1:sparc=)
89 INTEL_BLD_1= $(MACH:sparc=$(POUND_SIGN))
90 INTEL_BLD=   $(INTEL_BLD_1:i386=)
91 #
92 STRIP_COMMENTS= $(INTERNAL_RELEASE_BUILD)
93 #
94 # Are we building tonic closedbins? Unless you have used the
95 # -O flag to nightly or bldenv, leave the definition of TONICBUILD
96 # as $(POUND_SIGN).
97 #
98 # IF YOU CHANGE CLOSEDROOT, you MUST change install.bin
99 # to match the new definition.
100 TONICBUILD=   $(POUND_SIGN)
101 $(TONICBUILD)CLOSEDROOT= $(ROOT)-closed
102 #
103 #
104 # The variables below control the compilers used during the build.
105 # There are a number of permutations.
106 #
107 # __GNUC and __SUNC control (and indicate) the primary compiler. Whichever
108 # one is not POUND_SIGN is the primary, with the other as the shadow. They
109 # may also be used to control entirely compiler-specific Makefile assignments.
110 # __SUNC and Sun Studio are the default.
111 #
112 # __GNUC64 indicates that the 64bit build should use the GNU C compiler.
113 # There is no Sun C analogue.
114 #
115 # The following version-specific options are operative regardless of which
116 # compiler is primary, and control the versions of the given compilers to be
117 # used. They also allow compiler-version specific Makefile fragments.
118 #
119 #
120 __GNUC=              $(POUND_SIGN)
121 $(__GNUC)__SUNC=    $(POUND_SIGN)
122 __GNUC64=           $(__GNUC)
123 #
124 # CLOSED is the root of the tree that contains source which isn't released
125 # as open source
126 CLOSED=             $(SRC)/../closed
```

new/usr/src/Makefile.master

3

```

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

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

134 JAVA_ROOT=           /usr/java

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

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

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

147 RPCGEN=               /usr/bin/rpcgen
148 STABS=                $(ONBLD_TOOLS)/bin/$(MACH)/stabs
149 ELFXTRACT=            $(ONBLD_TOOLS)/bin/$(MACH)/elfextract
150 MBH_PATCH=            $(ONBLD_TOOLS)/bin/$(MACH)/mbh_patch
151 ECHO=                  echo
152 INS=                   install
153 TRUE=                  true
154 SYMLINK=              /usr/bin/ln -s
155 LN=                    /usr/bin/ln
156 CHMOD=                /usr/bin/chmod
157 MV=                    /usr/bin/mv -f
158 RM=                    /usr/bin/rm -f
159 CUT=                   /usr/bin/cut
160 NM=                    /usr/ccs/bin/nm
161 DIFF=                  /usr/bin/diff
162 GREP=                  /usr/bin/grep
163 EGREP=                 /usr/bin/egrep
164 ELFWRAP=               /usr/bin/elfwrap
165 KSH93=                 /usr/bin/ksh93
166 SED=                   /usr/bin/sed
167 AWK=                   /usr/xpg4/bin/awk
167 NAWK=                  /usr/bin/nawk
168 CP=                     /usr/bin/cp -f
169 MCS=                    /usr/ccs/bin/mcs
170 CAT=                   /usr/bin/cat
171 ELFDUMP=               /usr/ccs/bin/elfdump
172 M4=                     /usr/ccs/bin/m4
173 STRIP=                 /usr/ccs/bin/strip
174 LEX=                    /usr/ccs/bin/lex
175 FLEX=                  $(SFW_ROOT)/bin/flex
176 YACC=                   /usr/ccs/bin/yacc
177 CPP=                    /usr/lib/cpp
178 JAVAC=                 $(JAVA_ROOT)/bin/javac
179 JAVAH=                  $(JAVA_ROOT)/bin/javah
180 JAVADOC=                $(JAVA_ROOT)/bin/javadoc
181 RMIC=                   $(JAVA_ROOT)/bin/rmic
182 JAR=                    $(JAVA_ROOT)/bin/jar
183 CTFCONVERT=            $(ONBLD_TOOLS)/bin/$(MACH)/ctfconvert
184 CTFMERGE=              $(ONBLD_TOOLS)/bin/$(MACH)/ctfmerge
185 CTFSTABS=              $(ONBLD_TOOLS)/bin/$(MACH)/ctfstabs
186 CTFSTRIP=              $(ONBLD_TOOLS)/bin/$(MACH)/ctfstrip
187 NDRGEN=                 $(ONBLD_TOOLS)/bin/$(MACH)/ndrgen
188 GENOFFSETS=            $(ONBLD_TOOLS)/bin/genoffsets
189 CTFCVTPTBL=            $(ONBLD_TOOLS)/bin/ctfcvtptbl
190 CTFINDMOD=              $(ONBLD_TOOLS)/bin/ctffindmod
191 XREF=                   $(ONBLD_TOOLS)/bin/xref
192 FIND=                   /usr/bin/find

```

new/usr/src/Makefile.master

4

```

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

206 FILEMODE=               644
207 DIRMODE=                 755

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

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

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

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

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

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

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

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

```

```

259 # install a link
260 INSLINKTARGET= $<
261 INS.link= $(RM) $@; $(LN) $(INSLINKTARGET) $@
262 INS.symlink= $(RM) $@; $(SYMLINK) $(INSLINKTARGET) $@

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

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

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

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

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

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

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

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

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

321 # Prevent the removal of static symbols by the SPARC code generator (cg).
322 # The x86 code generator (ube) does not remove such symbols and as such
323 # using this workaround is not applicable for x86.
324 #

```

```

325 CCSTATICSYM= -Wc,-Qassembler-ounrefsym=0
326 #
327 # generate 32-bit addresses in the v9 kernel. Saves memory.
328 CCABS32= -Wc,-xcode=abs32
329 #
330 # generate v9 code which tolerates callers using the v7 ABI, for the sake of
331 # system calls.
332 CC32BITCALLERS= -_gcc=-massume-32bit-callers

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

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

358 #
359 # compiler '-xarch' flag. This is here to centralize it and make it
360 # overridable for testing.
361 sparc_XARCH= -m32
362 sparcv9_XARCH= -m64
363 i386_XARCH=
364 amd64_XARCH= -m64 -Ui386 -U_i386

366 # assembler '-xarch' flag. Different from compiler '-xarch' flag.
367 sparc_AS_XARCH= -xarch=v8plus
368 sparcv9_AS_XARCH= -xarch=v9
369 i386_AS_XARCH=
370 amd64_AS_XARCH= -xarch=amd64 -P -Ui386 -U_i386

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

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

390 STAND_FLAGS_32 = $($(_MACH)_STAND_FLAGS)

```



```

391 STAND_FLAGS_64 = $($ (MACH64)_STAND_FLAGS)

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

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

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

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

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

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

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

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

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

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

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

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

```

```

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

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

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

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

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

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

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

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

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

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

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

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

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

```

```

523 sparcv9_IROPTFLAG      = -W2,-xwrap_int
524 i386_IROPTFLAG        =
525 amd64_IROPTFLAG        =

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

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

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

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

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

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

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

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

588 sparc_CCFLAGS=        -cg92 -compat=4 \

```

```

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

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

610 CCOPTFLAG=           $($MACH)_CCOPTFLAG
611 CCOPTFLAG64=         $($MACH64)_CCOPTFLAG
612 CCFLAGS=             $(COPTFLAG) $($MACH)_CCFLAGS
613 CCFLAGS64=           $(COPTFLAG64) $($MACH64)_CCFLAGS

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

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

633 #
634 # Generated mapfiles that are compiler specific, and used throughout the
635 # build. These mapfiles are not delivered in /usr/lib/ld.
636 #
637 MAPFILE.NGB_sparc=   $(SRC)/common/mapfiles/gen/sparc_cc_map.noexglobs
638 $(__GNUCC64)MAPFILE.NGB_sparc= \
639                        $(SRC)/common/mapfiles/gen/sparc_gcc_map.noexglobs
640 MAPFILE.NGB_sparcv9= $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexglobs
641 $(__GNUCC64)MAPFILE.NGB_sparcv9= \
642                        $(SRC)/common/mapfiles/gen/sparcv9_gcc_map.noexglobs
643 MAPFILE.NGB_i386=    $(SRC)/common/mapfiles/gen/i386_cc_map.noexglobs
644 $(__GNUCC64)MAPFILE.NGB_i386= \
645                        $(SRC)/common/mapfiles/gen/i386_gcc_map.noexglobs
646 MAPFILE.NGB_amd64=   $(SRC)/common/mapfiles/gen/amd64_cc_map.noexglobs
647 $(__GNUCC64)MAPFILE.NGB_amd64= \
648                        $(SRC)/common/mapfiles/gen/amd64_gcc_map.noexglobs
649 MAPFILE.NGB =        $(MAPFILE.NGB_$(MACH))

651 #
652 # A generic interface mapfile name, used by various dynamic objects to define
653 # the interfaces and interposers the object must export.
654 #

```

```

655 MAPFILE.INT =          mapfile-intf

657 #
658 # LDLIBS32 can be set in the environment to override the following assignment.
659 # LDLIBS64 can be set to override the assignment made in Makefile.master.64.
660 # These environment settings make sure that no libraries are searched outside
661 # of the local workspace proto area:
662 #     LDLIBS32=-YP,$ROOT/lib:$ROOT/usr/lib
663 #     LDLIBS64=-YP,$ROOT/lib/$MACH64:$ROOT/usr/lib/$MACH64
664 #
665 LDLIBS32 =          $(ENVLDLIBS1) $(ENVLDLIBS2) $(ENVLDLIBS3)
666 LDLIBS.cmd =       $(LDLIBS32)
667 LDLIBS.lib =       $(LDLIBS32)
668 #
669 # Define compilation macros.
670 #
671 COMPILE.c=         $(CC) $(CFLAGS) $(CPPFLAGS) -c
672 COMPILE64.c=       $(CC) $(CFLAGS64) $(CPPFLAGS) -c
673 COMPILE.cc=        $(CCC) $(CCFLAGS) $(CPPFLAGS) -c
674 COMPILE64.cc=     $(CCC) $(CCFLAGS64) $(CPPFLAGS) -c
675 COMPILE.s=         $(AS) $(ASFLAGS) $(AS_CPPFLAGS)
676 COMPILE64.s=      $(AS) $(ASFLAGS) $(MACH64_AS_XARCH) $(AS_CPPFLAGS)
677 COMPILE.d=         $(DTRACE) -G -32
678 COMPILE64.d=      $(DTRACE) -G -64
679 COMPILE.b=         $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))
680 COMPILE64.b=      $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))

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

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

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

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

```

```

721 ALWAYS_LINT_DEFS += -erroff=E_INTEGRAL_CONST_EXP_EXPECTED
722 ALWAYS_LINT_DEFS += -erroff=E_PASS_INT_TO_SMALL_INT
723 ALWAYS_LINT_DEFS += -erroff=E_PTR_CONV_LOSES_BITS

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

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

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

743 # Define native compilation macros
744 #

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

753 # Till SS12ul formally becomes the NV CBE, LINT is hard
754 # coded to be picked up from the $SPRO_ROOT/sunstudio12.1/
755 # location. Impacted variables are sparc_LINT, sparcv9_LINT,
756 # i386_LINT, amd64_LINT.
757 # Reset them when SS12ul is rolled out.
758 #

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

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

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

```

```

787 $(__GNUC)i386_AS= $(ONBLD_TOOLS)/bin/$(MACH)/aw
788 i386_LD= /usr/ccs/bin/ld
789 i386_LINT= $(SPRO_ROOT)/sunstudio12.1/bin/lint

791 amd64_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
792 $(__GNUC64)amd64_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
793 amd64_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
794 $(__GNUC64)amd64_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
795 amd64_CPP= /usr/ccs/lib/cpp
796 amd64_AS= $(ONBLD_TOOLS)/bin/$(MACH)/aw
797 amd64_LD= /usr/ccs/bin/ld
798 amd64_LINT= $(SPRO_ROOT)/sunstudio12.1/bin/lint

800 NATIVECC= $($ (NATIVE_MACH)_CC)
801 NATIVECCC= $($ (NATIVE_MACH)_CCC)
802 NATIVECPP= $($ (NATIVE_MACH)_CPP)
803 NATIVEAS= $($ (NATIVE_MACH)_AS)
804 NATIVELD= $($ (NATIVE_MACH)_LD)
805 NATIVELINT= $($ (NATIVE_MACH)_LINT)

807 #
808 # Makefile.master.64 overrides these settings
809 #
810 CC= $(NATIVECC)
811 CCC= $(NATIVECCC)
812 CPP= $(NATIVECPP)
813 AS= $(NATIVEAS)
814 LD= $(NATIVELD)
815 LINT= $(NATIVELINT)

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

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

826 BDIRECT= -Bdirect
827 BDYNAMIC= -Bdynamic
828 BLOCAL= -Blocal
829 BNODIRECT= -Bnodirect
830 BREDUCE= -Breduce
831 BSTATIC= -Bstatic

833 ZDEFS= -zdefs
834 ZDIRECT= -zdirect
835 ZIGNORE= -zignore
836 ZINITFIRST= -zinitfirst
837 ZINTERPOSE= -zinterpose
838 ZLAZYLOAD= -zlazyload
839 ZLOADFLTR= -zloadfltr
840 ZMULDEFS= -zmuldefs
841 ZNODEFAULTLIB= -znodefaultlib
842 ZNODEFS= -znodefs
843 ZNODELETE= -znodelete
844 ZNODLOPEN= -znodlopen
845 ZNODUMP= -znodump
846 ZNOLAZYLOAD= -znolazyload
847 ZNOLDYNSYM= -znoldynsym
848 ZNORELOC= -znoreloc
849 ZNOVERSION= -znoversion
850 ZRECORD= -zrecord
851 ZREDLOCSYM= -zredlocsym
852 ZTEXT= -ztext

```

```

853 ZVERBOSE= -zverbose

855 GSHARED= -G
856 CCMT= -mt

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

861 sparc_C_PICFLAGS = -K pic
862 sparcv9_C_PICFLAGS = -K pic
863 i386_C_PICFLAGS = -K pic
864 amd64_C_PICFLAGS = -K pic
865 C_PICFLAGS = $($ (MACH)_C_PICFLAGS)
866 C_PICFLAGS64 = $($ (MACH64)_C_PICFLAGS)

868 sparc_C_BIGPICFLAGS = -K PIC
869 sparcv9_C_BIGPICFLAGS = -K PIC
870 i386_C_BIGPICFLAGS = -K PIC
871 amd64_C_BIGPICFLAGS = -K PIC
872 C_BIGPICFLAGS = $($ (MACH)_C_BIGPICFLAGS)
873 C_BIGPICFLAGS64 = $($ (MACH64)_C_BIGPICFLAGS)

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

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

886 #
887 # Default label for CTF sections
888 #
889 CTFCVTFLAGS= -i -L VERSION

891 #
892 # Override to pass module-specific flags to ctfmerge. Currently used
893 # only by krtld to turn on fuzzy matching.
894 #
895 CTFMRGFLAGS=

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

899 ELFSIGN_O= $(TRUE)
900 ELFSIGN_CRYPTO= $(ELFSIGN_O)
901 ELFSIGN_OBJECT= $(ELFSIGN_O)
902 $(EXPORT_RELEASE_BUILD)ELFSIGN_O = $(ELFSIGN)
903 $(EXPORT_RELEASE_BUILD)ELFSIGN_CFNAME = SUNWosnetCF
904 $(EXPORT_RELEASE_BUILD)ELFSIGN_KEY = \
905 $(CLOSED)/cmd/cmd-crypto/etc/keys/$(ELFSIGN_CFNAME)
906 $(EXPORT_RELEASE_BUILD)ELFSIGN_CERT= \
907 $(CLOSED)/cmd/cmd-crypto/etc/certs/$(ELFSIGN_CFNAME)
908 $(EXPORT_RELEASE_BUILD)ELFSIGN_SENAME = SUNWosnetSE
909 $(EXPORT_RELEASE_BUILD)ELFSIGN_SEKEY = \
910 $(CLOSED)/cmd/cmd-crypto/etc/keys/$(ELFSIGN_SENAME)
911 $(EXPORT_RELEASE_BUILD)ELFSIGN_SECERT= \
912 $(CLOSED)/cmd/cmd-crypto/etc/certs/$(ELFSIGN_SENAME)
913 $(EXPORT_RELEASE_BUILD)ELFSIGN_CRYPTO= $(ELFSIGN_O) sign \
914 $(ELFSIGN_FORMAT_OPTION) \
915 -k $(ELFSIGN_KEY) -c $(ELFSIGN_CERT) -e $@
916 $(EXPORT_RELEASE_BUILD)ELFSIGN_OBJECT= $(ELFSIGN_O) sign \
917 $(ELFSIGN_FORMAT_OPTION) \
918 -k $(ELFSIGN_SEKEY) -c $(ELFSIGN_SECERT) -e $@

```

```

920 # Rules (normally from make.rules) and macros which are used for post
921 # processing files. Normally, these do stripping of the comment section
922 # automatically.
923 #   RELEASE_CM:      Should be edited to reflect the release.
924 #   POST_PROCESS_O:  Post-processing for '.o' files.
925 #   POST_PROCESS_A:  Post-processing for '.a' files (currently null).
926 #   POST_PROCESS_SO: Post-processing for '.so' files.
927 #   POST_PROCESS:    Post-processing for executable files (no suffix).
928 # Note that these macros are not completely generalized as they are to be
929 # used with the file name to be processed following.
930 #
931 # It is left as an exercise to Release Engineering to embellish the generation
932 # of the release comment string.
933 #
934 #   If this is a standard development build:
935 #   compress the comment section (mcs -c)
936 #   add the standard comment (mcs -a $(RELEASE_CM))
937 #   add the development specific comment (mcs -a $(DEV_CM))
938 #
939 #   If this is an installation build:
940 #   delete the comment section (mcs -d)
941 #   add the standard comment (mcs -a $(RELEASE_CM))
942 #   add the development specific comment (mcs -a $(DEV_CM))
943 #
944 #   If this is an release build:
945 #   delete the comment section (mcs -d)
946 #   add the standard comment (mcs -a $(RELEASE_CM))
947 #
948 # The following list of macros are used in the definition of RELEASE_CM
949 # which is used to label all binaries in the build:
950 #
951 #   RELEASE          Specific release of the build, eg: 5.2
952 #   RELEASE_MAJOR    Major version number part of $(RELEASE)
953 #   RELEASE_MINOR    Minor version number part of $(RELEASE)
954 #   VERSION          Version of the build (alpha, beta, Generic)
955 #   PATCHID          If this is a patch this value should contain
956 #                   the patchid value (eg: "Generic 100832-01"), otherwise
957 #                   it will be set to $(VERSION)
958 #   RELEASE_DATE     Date of the Release Build
959 #   PATCH_DATE       Date the patch was created, if this is blank it
960 #                   will default to the RELEASE_DATE
961 #
962 #   RELEASE_MAJOR=   5
963 #   RELEASE_MINOR=   11
964 #   RELEASE=         $(RELEASE_MAJOR).$(RELEASE_MINOR)
965 #   VERSION=         SunOS Development
966 #   PATCHID=         $(VERSION)
967 #   RELEASE_DATE=    release date not set
968 #   PATCH_DATE=      $(RELEASE_DATE)
969 #   RELEASE_CM=      "@{$(POUND_SIGN)}SunOS $(RELEASE) $(PATCHID) $(PATCH_DATE)"
970 #   DEV_CM=          "@{$(POUND_SIGN)}SunOS Internal Development: non-nightly build"
971 #
972 #   PROCESS_COMMENT= @?${MCS} -c -a $(RELEASE_CM) -a $(DEV_CM)
973 #   $(STRIP_COMMENTS)PROCESS_COMMENT= @?${MCS} -d -a $(RELEASE_CM) -a $(DEV_CM)
974 #   $(RELEASE_BUILD)PROCESS_COMMENT= @?${MCS} -d -a $(RELEASE_CM)
975 #
976 #   STRIP_STABS=     :
977 #   $(RELEASE_BUILD)STRIP_STABS= $(STRIP) -x $@
978 #
979 #   POST_PROCESS_O= $(PROCESS_COMMENT) $@
980 #   POST_PROCESS_A=
981 #   POST_PROCESS_SO= $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
982 #                   $(ELFSIGN_OBJECT)
983 #   POST_PROCESS=    $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
984 #                   $(ELFSIGN_OBJECT)

```

```

986 #
987 #   chk4ubin is a tool that inspects a module for a symbol table
988 #   ELF section size which can trigger an OBP bug on older platforms.
989 #   This problem affects only specific sun4u bootable modules.
990 #
991 #   CHK4UBIN=        $(ONBLD_TOOLS)/bin/$(MACH)/chk4ubin
992 #   CHK4UBINFLAGS=
993 #   CHK4UBINARY=     $(CHK4UBIN) $(CHK4UBINFLAGS) $@
994 #
995 #
996 #   PKGARCHIVE specifies the default location where packages should be
997 #   placed if built.
998 #
999 #   $(RELEASE_BUILD)PKGARCHIVESUFFIX= -nd
1000 #   PKGARCHIVE=$(SRC)/../..../packages/$(MACH)/nightly$(PKGARCHIVESUFFIX)
1001 #
1002 #
1003 #   The repositories will be created with these publisher settings. To
1004 #   update an image to the resulting repositories, this must match the
1005 #   publisher name provided to "pkg set-publisher."
1006 #
1007 #   PKGPUBLISHER_REDIST= on-nightly
1008 #   PKGPUBLISHER_NONREDIST= on-extra
1009 #
1010 #   Default build rules which perform comment section post-processing.
1011 #
1012 #   .c:
1013 #   $(LINK.c) -o $@ $< $(LDLIBS)
1014 #   $(POST_PROCESS)
1015 #   .c.o:
1016 #   $(COMPILE.c) $(OUTPUT_OPTION) $< $(CTFCONVERT_HOOK)
1017 #   $(POST_PROCESS_O)
1018 #   .c.a:
1019 #   $(COMPILE.c) -o $% $<
1020 #   $(PROCESS_COMMENT) $%
1021 #   $(AR) $(ARFLAGS) $@ $%
1022 #   $(RM) $%
1023 #   .s.o:
1024 #   $(COMPILE.s) -o $@ $<
1025 #   $(POST_PROCESS_O)
1026 #   .s.a:
1027 #   $(COMPILE.s) -o $% $<
1028 #   $(PROCESS_COMMENT) $%
1029 #   $(AR) $(ARFLAGS) $@ $%
1030 #   $(RM) $%
1031 #   .cc:
1032 #   $(LINK.cc) -o $@ $< $(LDLIBS)
1033 #   $(POST_PROCESS)
1034 #   .cc.o:
1035 #   $(COMPILE.cc) $(OUTPUT_OPTION) $<
1036 #   $(POST_PROCESS_O)
1037 #   .cc.a:
1038 #   $(COMPILE.cc) -o $% $<
1039 #   $(AR) $(ARFLAGS) $@ $%
1040 #   $(PROCESS_COMMENT) $%
1041 #   $(RM) $%
1042 #   .y:
1043 #   $(YACC.y) $<
1044 #   $(LINK.c) -o $@ y.tab.c $(LDLIBS)
1045 #   $(POST_PROCESS)
1046 #   $(RM) y.tab.c
1047 #   .y.o:
1048 #   $(YACC.y) $<
1049 #   $(COMPILE.c) -o $@ y.tab.c $(CTFCONVERT_HOOK)
1050 #   $(POST_PROCESS_O)

```

```

1051      $(RM) y.tab.c
1052 .l:
1053      $(RM) $*.c
1054      $(LEX.l) $< > $*.c
1055      $(LINK.c) -o $@ $*.c -ll $(LDLIBS)
1056      $(POST_PROCESS)
1057      $(RM) $*.c
1058 .l.o:
1059      $(RM) $*.c
1060      $(LEX.l) $< > $*.c
1061      $(COMPILE.c) -o $@ $*.c $(CTFCONVERT_HOOK)
1062      $(POST_PROCESS_O)
1063      $(RM) $*.c

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

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

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

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

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

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

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

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

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

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

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

```

```

1117 # causing xgettext to put all messages into the default domain.
1118 #
1119 CPPFORPO=$(COMPILE.cpp:\ "$(TEXT_DOMAIN)\ "=TEXT_DOMAIN)

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

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

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

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

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

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

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

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

1158 .h.check:
1159      $(DOT_H_CHECK)

1161 .x.check:
1162      $(DOT_X_CHECK)

1164 .xml.xmlchk:
1165      $(MANIFEST_CHECK)

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

1174 #
1175 # Include rules to render automated sccs get rules "safe".
1176 #
1177 include $(SRC)/Makefile.noget

```

new/usr/src/Makefile.msg.targ

1

```

*****
2271 Sun May 26 18:22:32 2013
new/usr/src/Makefile.msg.targ
XXX Remove nawk(1)
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2010 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # Makefile.msg.targ
26 #

28 #
29 # This file contains common targets for building message catalogues.
30 # Over time, existing Makefiles should be rewritten to include this file
31 # rather than use their own private copy of these rules.
32 #

34 .PARALLEL:      $(POFILES) $(MSGFILES)

36 $(MSGDOMAIN)% $(DCMSGDOMAIN)%:
37     $(INS.dir)

39 $(MSGDOMAINPOFILE): $(MSGDOMAIN) $(POFILE)
40     $(RM) $@; $(CP) $(POFILE) $@

42 $(DCMSGDOMAINPOFILE): $(DCMSGDOMAIN) $(DCFILE)
43     $(RM) $@; $(CP) $(DCFILE) $@

45 BUILDPO.msgfiles = \
46     $(RM) messages.po $(TEXT_DOMAIN).po; \
47     $(TOUCH) $(TEXT_DOMAIN).po; \
48     $(XGETTEXT) $(XGETTEXTFLAGS) $(MSGFILES); \
49     $(SED) -e '/^\# msgid/,/^\# msgstr/d' -e '/^domain/d' \
50           -e '/^\#\$\$/d' -e '/^\#/, /d' \
51           messages.po $(TEXT_DOMAIN).po > $(POFILE); \
52     $(RM) messages.po $(TEXT_DOMAIN).po

54 BUILDPO.pofiles = \
55     $(RM) $(POFILE); \
56     $(AWK) '\
57     $(NAWK) '\
58     FNR == 1 { print "\# Messages from " FILENAME; } \
59     /^domain/ { next; } \
60     /^msgid/, /^msgstr/ { msg = msg "\n" $$0; \
        if ( $$0 !~ /^msgstr/ ) next; \

```

new/usr/src/Makefile.msg.targ

2

```

61     if (msg in messages) { \
62         gsub(/\n/, "\n\#", msg); \
63     } else { \
64         messages[msg] = 1; \
65     } \
66     sub(/^\n/, "", msg); \
67     print msg; \
68     msg = ""; \
69     next; \
70     } \
71     /^#\ [\.]*Messages/ { \
72         sub(/Messages/, "...Messages"); } \
73     { print; }' $(POFILES) > $(POFILE)

74 pofile_MSGFILES:
75     $(BUILDPO.msgfiles)

77 pofile_POFILES:
78     $(BUILDPO.pofiles)

```

```

*****
11442 Sun May 26 18:22:32 2013
new/usr/src/cmd/Makefile
XXX Remove awk(1)
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright 2010 Nexenta Systems, Inc. All rights reserved.
24 # Copyright 2011 Joyent, Inc. All rights reserved.
25 # Copyright (c) 2012 by Delphix. All rights reserved.
26 # Copyright (c) 2013 DEY Storage Systems, Inc. All rights reserved.
27 #
28 include ../Makefile.master
29 #
30 #
31 # Note that the commands 'agents', 'lp', 'perl', and 'man' are first in
32 # the list, violating alphabetical order. This is because they are very
33 # long-running and should be given the most wall-clock time for a
34 # parallel build.
35 #
36 # Commands in the FIRST_SUBDIRS list are built before starting the build
37 # of other commands. Currently this includes only 'isaexec' and
38 # 'platexec'. This is necessary because $(ROOT)/usr/lib/isaexec or
39 # $(ROOT)/usr/lib/platexec must exist when some other commands are built
40 # because their 'make install' creates a hard link to one of them.
41 #
42 # Commands are listed one per line so that TeamWare can auto-merge most
43 # changes.
44 #
45 #
46 FIRST_SUBDIRS= \
47 isaexec \
48 platexec
49 #
50 COMMON_SUBDIRS= \
51 allocate \
52 availdevs \
53 lp \
54 perl \
55 man \
56 Adm \
57 abi \
58 adbgen \
59 acct \
60 acctadm \
61 arch

```

```

62 asa \
63 ast \
64 audio \
65 auths \
66 autopush \
67 avs \
68 awk \
69 awk_xpg4 \
70 backup \
71 banner \
72 bart \
73 basename \
74 bc \
75 bdiff \
76 beadm \
77 bfs \
78 bnu \
79 boot \
80 busstat \
81 cal \
82 calendar \
83 captinfo \
84 cat \
85 cdrw \
86 cfgadm \
87 checkeq \
88 checknr \
89 chgrp \
90 chmod \
91 chown \
92 chroot \
93 clear \
94 clinfo \
95 cmd-crypto \
96 cmd-inet \
97 col \
98 compress \
99 consadm \
100 coreadm \
101 cpio \
102 cpc \
103 cron \
104 crypt \
105 csh \
106 csplit \
107 ctrun \
108 ctstat \
109 ctwatch \
110 datadm \
111 date \
112 dc \
113 dd \
114 deroff \
115 devfsadm \
116 syseventd \
117 devctl \
118 devinfo \
119 devmgmt \
120 devprop \
121 dfs.cmds \
122 diff \
123 diff3 \
124 diffmk \
125 dircmp \
126 dirname \
127 dis \

```


new/usr/src/cmd/Makefile

```

127      diskmgtd      //
128      dispadmin    //
129      dladm         //
130      dlstat       //
131      dmesg        //
132      dodatadm     //
133      dtrace       //
134      du           //
135      dumpadm     //
136      dumpcs      //
137      echo        //
138      ed          //
139      eeprom      //
140      egrep       //
141      eject       //
142      emul64ioctl //
143      enhance     //
144      env         //
145      eqn         //
146      expand      //
147      expr       //
148      exstr      //
149      factor     //
150      false      //
151      fcinfo     //
152      fcoesvc    //
153      fdetach    //
154      fdformat   //
155      fdisk      //
156      filesync   //
157      fgrep      //
158      file       //
159      filebench  //
160      find       //
161      flowadm   //
162      flowstat  //
163      fm        //
164      fmt       //
165      fmthard  //
166      fmsg     //
167      fold     //
168      format   //
169      fs.d     //
170      fstyp   //
171      fuser   //
172      fwflash //
173      gcore   //
174      gencat  //
175      geniconvtbl //
176      genmsg  //
177      getconf //
178      getdevpolicy //
179      getent  //
180      getfacl //
181      getmajor //
182      getopt //
183      gettext //
184      gettxt  //
185      grep   //
186      grep_xpg4 //
187      groups //
188      grpck  //
189      gss    //
190      hal   //
191      halt  //
192      head  //

```

3

new/usr/src/cmd/Makefile

```

193      hostid      //
194      hostname   //
195      hotplug    //
196      hotplugd   //
197      hwdata     //
198      ibd_upgrade //
199      id         //
200      idmap      //
201      infocmp    //
202      init       //
203      initpkg   //
204      install.d //
205      intrd     //
206      intrstat  //
207      ipcrm     //
208      ipcs      //
209      ipf       //
210      isainfo   //
211      isalist   //
212      itutools  //
213      iscsiadm  //
214      iscsid    //
215      iscsitsvc //
216      isns     //
217      itadm    //
218      java     //
219      kbd      //
220      keyserv  //
221      killall //
222      krb5     //
223      ksh     //
224      kvmstat //
225      last    //
226      lastcomm //
227      latencytop //
228      ldap    //
229      ldapcachemgr //
230      lgrpinfo //
231      line    //
232      link    //
233      dlmgmt //
234      listen //
235      loadkeys //
236      locale //
237      localedef //
238      lockstat //
239      locator //
240      lofiadm //
241      logadm  //
242      logger  //
243      login   //
244      logins  //
245      look    //
246      ls      //
247      luxadm  //
248      lvm     //
249      mach    //
250      machid //
251      mail    //
252      mailx  //
253      makekey //
254      mdb     //
255      msg     //
256      mkdir  //
257      mkfifo //
258      mkfile //

```

4

new/usr/src/cmd/Makefile

259 mkmsgs //
260 mknod //
261 mkpwdict //
262 mktemp //
263 modload //
264 more //
265 mpathadm //
266 msgfmt //
267 msgid //
268 mt //
269 mv //
270 mvdir //
271 ndmpadm //
272 ndmpd //
273 ndmpstat //
274 netadm //
275 netfiles //
276 newform //
277 newgrp //
278 news //
279 newtask //
280 nice //
281 nl //
282 nlsadmin //
283 nohup //
284 nsadmin //
285 nscd //
286 oamuser //
287 oawk //
288 od //
289 pack //
290 pagesize //
291 passmgmt //
292 passwd //
293 pathchk //
294 pbind //
295 pcidr //
296 pcitool //
297 pfexec //
298 pfexecd //
299 pginfo //
300 pgstat //
301 pgrep //
302 picl //
303 plimit //
304 policykit //
305 pools //
306 power //
307 powertop //
308 ppgsz //
309 pg //
310 plockstat //
311 pr //
312 prctl //
313 print //
314 printf //
315 priocntl //
316 profiles //
317 projadd //
318 projects //
319 prstat //
320 prtconf //
321 prtdiag //
322 prtvtoc //
323 ps //
324 psradm //

new/usr/src/cmd/Makefile

325 psrinfo //
326 psrset //
327 ptools //
328 pwck //
329 pwconv //
330 pwd //
331 pyzfs //
332 raidctl //
333 ramdiskadm //
334 rcap //
335 rcm_daemon //
336 rctladm //
337 refer //
338 regcmp //
339 renice //
340 rexd //
341 rm //
342 rmdir //
343 rmformat //
344 rmmount //
345 rmt //
346 rmvolmgr //
347 roles //
348 rpcbind //
349 rpcgen //
350 rpcinfo //
351 rpcsvc //
352 runat //
353 sa //
354 saf //
355 sasinfo //
356 savecore //
357 sbdadm //
358 script //
359 scsi //
360 sdiff //
361 sdpadm //
362 sed //
363 sendmail //
364 setfacl //
365 setmnt //
366 setpgrp //
367 setuname //
368 sgs //
369 sh //
370 shcomp //
371 sbios //
372 smbsrv //
373 smserverd //
374 soelim //
375 sort //
376 spell //
377 split //
378 sqlite //
379 srchtxt //
380 srptadm //
381 srptsvc //
382 ssh //
383 stat //
384 stmfadm //
385 stmfproxy //
386 stmfsvc //
387 stmsboot //
388 streams //
389 strings //
390 su //

new/usr/src/cmd/Makefile

```

391     sulin          \
392     sunpc         \
393     svc           \
394     svr4pkg      \
395     swap         \
396     sync         \
397     sysdef       \
398     syseventadm  \
399     syslogd      \
400     tabs         \
401     tail         \
402     tar          \
403     tbl          \
404     tcopy        \
405     tcpd         \
406     terminfo     \
407     th_tools     \
408     tic          \
409     time         \
410     tip          \
411     tnf          \
412     touch        \
413     tput         \
414     tr           \
415     trapstat    \
416     troff        \
417     true         \
418     truss        \
419     tsol         \
420     tty          \
421     ttymon       \
422     tzreload     \
423     uadmin       \
424     ul           \
425     uname        \
426     units        \
427     unlink       \
428     unpack       \
429     userattr     \
430     users        \
431     utmp_update  \
432     utmpd        \
433     valtools     \
434     vgrind       \
435     vi           \
436     volcheck     \
437     volrmmount  \
438     vrrpadm      \
439     vscan        \
440     vt           \
441     w            \
442     wall         \
443     which        \
444     who          \
445     whodo        \
446     wracct       \
447     write        \
448     wusbadm      \
449     xargs        \
450     xstr         \
451     yes          \
452     ypcmd        \
453     yppasswd     \
454     zdb          \
455     zdump        \
456     zfs          \

```

7

new/usr/src/cmd/Makefile

```

457     zhack        \
458     zic          \
459     zinject      \
460     zlogin       \
461     zoneadm      \
462     zoneadmd     \
463     zonecfg      \
464     zonename     \
465     zpool        \
466     zlook        \
467     zonestat     \
468     zstreamdump  \
469     ztest        \

471 $(CLOSED_BUILD)COMMON_SUBDIRS += \
472     $(CLOSED)/cmd/iconv \
473     $(CLOSED)/cmd/ksh \
474     $(CLOSED)/cmd/localedef \
475     $(CLOSED)/cmd/more_xpg4 \
476     $(CLOSED)/cmd/mtst \
477     $(CLOSED)/cmd/od \
478     $(CLOSED)/cmd/patch \
479     $(CLOSED)/cmd/pax \
480     $(CLOSED)/cmd/printf \
481     $(CLOSED)/cmd/sed \
482     $(CLOSED)/cmd/sed_xpg4 \

484 i386_SUBDIRS= \
485     acpihpd      \
486     addbadsec    \
487     biosdev      \
488     diskscan     \
489     lms          \
490     ntfsprogs    \
491     parted       \
492     rtc          \
493     ucodeadm     \
494     xvm          \

496 sparc_SUBDIRS= \
497     cvcd         \
498     dcs          \
499     device_remap \
500     drd          \
501     fruadm       \
502     ldmd         \
503     oplhpd       \
504     prtdscp      \
505     prtfru       \
506     scadm        \
507     sckmd        \
508     sf880drd     \
509     virtinfo     \
510     vntsd        \

512 #
513 # Commands that are messaged. Note that 'lp' and 'man' come first
514 # (see previous comment about 'lp' and 'man').
515 #
516 MSGSUBDIRS= \
517     lp           \
518     man          \
519     abi          \
520     acctadm      \
521     allocate     \
522     asa          \

```

8

```

523 audio //
524 audit //
525 auditconfig //
526 auditd //
527 auditrecord //
528 auditset //
529 auths //
530 autopush //
531 avs //
533 awk //
532 awk_xpg4 //
533 backup //
534 banner //
535 bart //
536 basename //
537 beadm //
538 bnu //
539 busstat //
540 cal //
541 cat //
542 cdrw //
543 cfgadm //
544 checkeq //
545 checknr //
546 chgrp //
547 chmod //
548 chown //
549 cmd-crypto //
550 cmd-inet //
551 col //
552 compress //
553 consadm //
554 coreadm //
555 cpio //
556 cpc //
557 cron //
558 csh //
559 csplit //
560 ctrun //
561 ctstat //
562 ctwatch //
563 datadm //
564 date //
565 dc //
566 dcs //
567 dd //
568 deroff //
569 devfsadm //
570 dfs.cmds //
571 diff //
572 diffmk //
573 dladm //
574 dlstat //
575 du //
576 dumpcs //
577 ed //
578 eject //
579 env //
580 eqn //
581 expand //
582 expr //
583 fcinfo //
584 fgrep //
585 file //
586 filesync //
587 find //

```

```

588 flowadm //
589 flowstat //
590 fm //
591 fold //
592 fs.d //
593 fwflash //
594 geniconvtbl //
595 genmsg //
596 getconf //
597 getent //
598 gettext //
599 gettxt //
600 grep //
601 grep_xpg4 //
602 grpck //
603 gss //
604 halt //
605 head //
606 hostname //
607 hotplug //
608 id //
609 idmap //
610 isaexec //
611 iscsiadm //
612 iscsid //
613 isns //
614 itadm //
615 kbd //
616 krb5 //
617 ksh //
618 last //
619 ldap //
620 ldapcachemgr //
621 lgrpinfo //
622 locale //
623 lofiadm //
624 logadm //
625 logger //
626 logins //
627 ls //
628 luxadm //
629 lvm //
630 mailx //
631 msg //
632 mkdir //
633 mkpdict //
634 mktemp //
635 more //
636 mpathadm //
637 msgfmt //
638 mv //
639 ndmpadm //
640 ndmpstat //
641 newgrp //
642 newtask //
643 nice //
644 nohup //
645 oawk //
646 pack //
647 passwd //
648 passmgmt //
649 pathchk //
650 pexec //
651 pg //
652 pgrep //
653 picl //

```

```

654     pools           \
655     power           \
656     pr              \
657     praudit        \
658     print           \
659     profiles        \
660     projadd         \
661     projects        \
662     prstat         \
663     prtdiag        \
664     ps              \
665     psrinfo         \
666     ptools          \
667     pwconv          \
668     pwd             \
669     pyzfs           \
670     raidctl         \
671     ramdiskadm     \
672     rcap            \
673     rcm_daemon     \
674     refer           \
675     regcmp          \
676     renice          \
677     roles           \
678     rm              \
679     rmdir           \
680     rmformat        \
681     rmmount         \
682     rmvolmgr       \
683     sasinfo         \
684     sbdadm          \
685     scadm           \
686     script          \
687     scsi            \
688     sdiff           \
689     sdpadm          \
690     sgs             \
691     sh              \
692     shcomp          \
693     smbshr          \
694     sort            \
695     split           \
696     srptadm         \
697     ssh             \
698     stat            \
699     stmfadm         \
700     stmsboot       \
701     strings         \
702     su              \
703     svc             \
704     svr4pkg        \
705     swap            \
706     syseventadm    \
707     syseventd      \
708     tabs            \
709     tar             \
710     tbl             \
711     time           \
712     tnf             \
713     touch           \
714     tput            \
715     troff           \
716     tsol            \
717     tty            \
718     ttymon         \
719     tzreload       \

```

```

720     ul              \
721     uname           \
722     units           \
723     unlink          \
724     unpack          \
725     userattr        \
726     valtools        \
727     vgrind          \
728     vi              \
729     volcheck        \
730     volrmmount     \
731     vrrpadm         \
732     vscan           \
733     w               \
734     who             \
735     whodo           \
736     wracct          \
737     write           \
738     wusbadm         \
739     xargs           \
740     yppasswd        \
741     zdump           \
742     zfs             \
743     zic             \
744     zlogin          \
745     zoneadm         \
746     zoneadm        \
747     zonecfg         \
748     zonename        \
749     zpool           \
750     zonestat        \
752 $(CLOSED_BUILD)MSGSUBDIRS += \
753     $(CLOSED)/cmd/iconv \
754     $(CLOSED)/cmd/ksh   \
755     $(CLOSED)/cmd/localedef \
756     $(CLOSED)/cmd/more_xpg4 \
757     $(CLOSED)/cmd/od    \
758     $(CLOSED)/cmd/patch \
759     $(CLOSED)/cmd/pax   \
760     $(CLOSED)/cmd/printf \
761     $(CLOSED)/cmd/sed   \
762     $(CLOSED)/cmd/sed_xpg4 \
764 sparc_MSGSUBDIRS= \
765     fruadm           \
766     prtdscp         \
767     prtfru          \
768     virtinfo        \
769     vntsd           \
771 i386_MSGSUBDIRS= \
772     ucodeadm        \
774 # \
775 # commands that use dcgettext for localized time, LC_TIME \
776 # \
777 DCSUBDIRS= \
778     cal              \
779     cfgadm           \
780     diff             \
781     ls               \
782     pr               \
783     ps               \
784     tar              \
785     w                \

```

new/usr/src/cmd/Makefile

```

786     who           \
787     whodo        \
788     write

790 $(CLOSED_BUILD)DCSUBDIRS += \
791     $(CLOSED)/cmd/pax

793 #
794 # commands that belong only to audit.
795 #
796 AUDITSUBDIRS=      \
797     amt             \
798     audit           \
799     audit_warn     \
800     auditconfig    \
801     auditd         \
802     auditrecord    \
803     auditreduce    \
804     auditset       \
805     auditstat      \
806     praudit

808 #
809 # commands not owned by the systems group
810 #
811 BWOSDIRS=

814 all :=            TARGET = all
815 install :=       TARGET = install
816 clean :=         TARGET = clean
817 clobber :=       TARGET = clobber
818 lint :=          TARGET = lint
819 _msg :=          TARGET = _msg
820 _dc :=           TARGET = _dc

822 .KEEP_STATE:

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

826 .PARALLEL:      $(BWOSDIRS) $(SUBDIRS) $(MSGSUBDIRS) $(AUDITSUBDIRS)

828 all install clean clobber lint: $(FIRST_SUBDIRS) .WAIT $(SUBDIRS) \
829     $(AUDITSUBDIRS)

831 #
832 # Manifests cannot be checked in parallel, because we are using
833 # the global repository that is in $(SRC)/cmd/svc/seed/global.db.
834 # For this reason, to avoid .PARALLEL and .NO_PARALLEL conflicts,
835 # we spawn off a sub-make to perform the non-parallel 'make check'
836 #
837 check:
838     $(MAKE) -f Makefile.check check

840 #
841 # The .WAIT directive works around an apparent bug in parallel make.
842 # Evidently make was getting the target _msg vs. _dc confused under
843 # some level of parallelization, causing some of the _dc objects
844 # not to be built.
845 #
846 _msg: $(MSGSUBDIRS) $($ (MACH)_MSGSUBDIRS) .WAIT _dc

848 _dc: $(DCSUBDIRS)

850 #
851 # Dependencies

```

13

new/usr/src/cmd/Makefile

```

852 #
853 fs.d: fstyp
854 ksh:  shcomp isaexec
855 mdb:  terminfo
856 print: lp

858 $(FIRST_SUBDIRS) $(BWOSDIRS) $(SUBDIRS) $(AUDITSUBDIRS): FRC
859     @if [ -f $@/Makefile ]; then \
860         cd $@; pwd; $(MAKE) $(TARGET); \
861     else \
862         true; \
863     fi

865 FRC:

```

14

new/usr/src/cmd/acct/lastlogin.sh

1

```
*****
2829 Sun May 26 18:22:33 2013
new/usr/src/cmd/acct/lastlogin.sh
XXX Remove awk(1)
*****
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License, Version 1.0 only
7 # (the "License"). You may not use this file except in compliance
8 # with the License.
9 #
10 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
11 # or http://www.opensolaris.org/os/licensing.
12 # See the License for the specific language governing permissions
13 # and limitations under the License.
14 #
15 # When distributing Covered Code, include this CDDL HEADER in each
16 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
17 # If applicable, add the following below this CDDL HEADER, with the
18 # fields enclosed by brackets "[]" replaced with your own identifying
19 # information: Portions Copyright [yyyy] [name of copyright owner]
20 #
21 # CDDL HEADER END
22 #
23 # Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T
24 # All Rights Reserved
```

```
27 #ident "%Z%M% %I% %E% SMI" /* SVr4.0 1.6 */
27 # "lastlogin - keep record of date each person last logged in"
28 # "bug - the date shown is usually 1 more than it should be"
29 # " because lastlogin is run at 4am and checks the last"
30 # " 24 hrs worth of process accounting info (in pacct)"
31 PATH=/usr/lib/acct:/usr/bin:/usr/sbin
32 cd /var/adm/acct
33 if test ! -r sum/loginlog; then
34 nulladm sum/loginlog
35 fi
36 # "cleanup loginlog - delete entries of those no longer in"
37 # "/etc/passwd and add an entry for those recently added"
38 # "line 1 - get file of current logins in same form as loginlog"
39 # "line 2 - merge the 2 files; use uniq to delete common"
40 # "lines resulting in those lines which need to be"
41 # "deleted or added from loginlog"
42 # "line 3 - result of sort will be a file with 2 copies"
43 # "of lines to delete and 1 copy of lines that are"
44 # "valid; use uniq to remove duplicate lines"
45 getent passwd | sed "s/\([^:]*\).*\/00-00-00 \1/" |\
46 sort +1 - sum/loginlog | uniq -u +10 |\
47 sort +1 - sum/loginlog | uniq -u > sum/tmploginlog
48 cp sum/tmploginlog sum/loginlog
49 # "update loginlog"
50 _d="date +%y-%m-%d"
51 _day="date +%m%d"
52 # "lines 1 and 2 - remove everything from the total"
53 # "acctng records with connect info except login"
54 # "name and adds the date"
55 # "line 3 - sorts in reverse order by login name; gets"
56 # "1st occurrence of each login name and resorts by date"
57 acctmerg -a < nite/ctacct.$_day | \
58 sed -e "s/^[^ ]*[ ]\([^\ ]*\)[ ].*$_d \1/" | \
59 /usr/xpg4/bin/awk '/^00-00-00/ {
60 awk '/^00-00-00/ {
```

new/usr/src/cmd/acct/lastlogin.sh

2

```
60 $0 = "00" $0
61 }
62 /^[0-9][0-9]-/ {
63 d=substr($0,1,2);
64 if (d<=68) {
65 $0 = "20" $0
66 } else {
67 $0 = "19" $0
68 }
69 }
70 { print }' - sum/loginlog | \
71 sort -r +1 | uniq +10 | sort | \
72 /usr/xpg4/bin/awk '/^[0-9][0-9][0-9][0-9]-/ {
73 awk '/^[0-9][0-9][0-9][0-9]-/ {
74 $0 = substr($0,3)
75 }
76 { print }' > sum/tmploginlog
77 cp sum/tmploginlog sum/loginlog
78 rm -f sum/tmploginlog
```

new/usr/src/cmd/allocate/disk_clean.sh

1

14110 Sun May 26 18:22:33 2013

new/usr/src/cmd/allocate/disk_clean.sh

XXX Remove awk(1)

unchanged portion omitted

```
220 #
221 # Find all mountpoints in use for a set of device special files.
222 # Usage: findmounts devpath ...
223 #

225 findmounts() {
226     /usr/xpg4/bin/awk -f - -v vold_root="$VOLD_ROOT" -v devs="$*" /etc/mntta
226     awk -f - -v vold_root="$VOLD_ROOT" -v devs="$*" /etc/mnttab <<\
227     "ENDOFAWKPGM"
228     BEGIN {
229         split(devs, devlist, " ");
230         for (devN in devlist) {
231             dev = devlist[devN];
232             realdevlist[dev] = 1;
233             sub(/.*\\/, "", dev);
234             sub(/s[0-9]$/, "", dev);
235             if (vold_root != "") {
236                 vold_dir[vold_root "/dev/dsk/" dev] = 1;
237                 vold_dir[vold_root "/dev/rdisk/" dev] = 1;
238             }
239         }
240     }

242     {
243         for (dev in realdevlist) {
244             if ($1 == dev) {
245                 mountpoint = $2;
246                 print mountpoint;
247             }
248         }
249         for (dev in vold_dir) {
250             if (substr($1, 1, length(dev)) == dev) {
251                 mountpoint = $2;
252                 print mountpoint;
253             }
254         }
255     }
256 ENDOFAWKPGM
257 }
```

unchanged portion omitted


```
new/usr/src/cmd/boot/scripts/create_diskmap.ksh
```

1

```
*****
```

```
1393 Sun May 26 18:22:35 2013
```

```
new/usr/src/cmd/boot/scripts/create_diskmap.ksh
```

```
XXX Remove awk(1)
```

```
*****
```

```
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License, Version 1.0 only
7 # (the "License"). You may not use this file except in compliance
8 # with the License.
9 #
10 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
11 # or http://www.opensolaris.org/os/licensing.
12 # See the License for the specific language governing permissions
13 # and limitations under the License.
14 #
15 # When distributing Covered Code, include this CDDL HEADER in each
16 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
17 # If applicable, add the following below this CDDL HEADER, with the
18 # fields enclosed by brackets "[]" replaced with your own identifying
19 # information: Portions Copyright [yyyy] [name of copyright owner]
20 #
21 # CDDL HEADER END
22 #
23 #
24 # Copyright 2005 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 #pragma ident      "%Z%M% %I%      %E% SMI"
28 #
29 #
30 GRUBDISKMAP=/var/run/solaris_grubdisk.map
31 #
32 /bin/rm -f "$GRUBDISKMAP"
33 /bin/touch "$GRUBDISKMAP"
34 /sbin/biosdev 2> /dev/null | while read diskno diskpath
35 do
36     devname=`/bin/ls -l /dev/rdisk/*p0 | /bin/grep $diskpath | /usr/xpg4/bin/
37     devname=`/bin/ls -l /dev/rdisk/*p0 | /bin/grep $diskpath | /bin/awk '{ p
38     ctdname=`echo $devname | /bin/sed "s#/dev/rdisk/##" | /bin/sed "s#p0##" `
39     grubdisk=`echo $diskno | /bin/sed "s/0x8/" `
40     echo "$grubdisk $ctdname $diskpath" >> "$GRUBDISKMAP"
41 done
42 # cleanup
```

```

new/usr/src/cmd/boot/scripts/create_ramdisk.ksh                                1
*****
14739 Sun May 26 18:22:36 2013
new/usr/src/cmd/boot/scripts/create_ramdisk.ksh
XXX Remove awk(1)
*****
_____unchanged_portion_omitted_____

157 function getsize
158 {
159     # Estimate image size and add 10% overhead for ufs stuff.
160     # Note, we can't use du here in case we're on a filesystem, e.g. zfs,
161     # in which the disk usage is less than the sum of the file sizes.
162     # The awk code
162     # The awk code
163     #
164     #         {t += ($5 % 1024) ? (int($5 / 1024) + 1) * 1024 : $5}
165     #
166     # below rounds up the size of a file/directory, in bytes, to the
167     # next multiple of 1024. This mimics the behavior of ufs especially
168     # with directories. This results in a total size that's slightly
169     # bigger than if du was called on a ufs directory.
170     size32=$(cat "$list32" | xargs -I {} ls -lLd "{}" 2> /dev/null |
171     /usr/xpg4/bin/awk '{t += ($5 % 1024) ? (int($5 / 1024) + 1) * 10
171     awk '{t += ($5 % 1024) ? (int($5 / 1024) + 1) * 1024 : $5}
172     END {print int(t * 1.10 / 1024)}')
173     (( size32 += dirsize32 ))
174     size64=$(cat "$list64" | xargs -I {} ls -lLd "{}" 2> /dev/null |
175     /usr/xpg4/bin/awk '{t += ($5 % 1024) ? (int($5 / 1024) + 1) * 10
175     awk '{t += ($5 % 1024) ? (int($5 / 1024) + 1) * 1024 : $5}
176     END {print int(t * 1.10 / 1024)}')
177     (( size64 += dirsize64 ))
178     (( total_size = size32 + size64 ))

180     if [ $compress = yes ] ; then
181         total_size=$(echo $total_size | /usr/xpg4/bin/awk '{print int($1
181         total_size=$(echo $total_size | awk '{print int($1 / 2)}')
182     fi
183 }
_____unchanged_portion_omitted_____

382 function create_archive
383 {
384     which=$1
385     archive=$2
386     lofidev=$3

388     echo "updating $archive"

390     if [ "$format" = "ufs" ] ; then
391         create_ufs "$which" "$archive" "$lofidev"
392     else
393         create_isofs "$which" "$archive"
394     fi

396     # sanity check the archive before moving it into place
397     #
398     ARCHIVE_SIZE=$(ls -l "${archive}-new" 2> /dev/null | /usr/xpg4/bin/awk '{
398     ARCHIVE_SIZE=$(ls -l "${archive}-new" 2> /dev/null | awk '{ print $5 }')
399     if [ $compress = yes ] || [ $ISA = sparc ] ; then
400         #
401         # 'file' will report "English text" for uncompressed
402         # boot_archives. Checking for that doesn't seem stable,
403         # so we just check that the file exists.
404         #
405         ls "${archive}-new" >/dev/null 2>&1
406     else

```

```

new/usr/src/cmd/boot/scripts/create_ramdisk.ksh                                2
407     #
408     # the file type check also establishes that the
409     # file exists at all
410     #
411     LC_MESSAGES=C file "${archive}-new" | grep gzip > /dev/null
412     fi

414     if [ $? = 1 ] && [ -x $GZIP_CMD ] || [ "$ARCHIVE_SIZE" -lt 10000 ]
415     then
416         #
417         # Two of these functions may be run in parallel. We
418         # need to allow the other to clean up, so we can't
419         # exit immediately. Instead, we set a flag.
420         #
421         echo "update of $archive failed"
422         ERROR=1
423     else
424         lockfs -f "/$ALT_ROOT" 2>/dev/null
425         mv "${archive}-new" "$archive"
426         lockfs -f "/$ALT_ROOT" 2>/dev/null
427     fi

429 }
_____unchanged_portion_omitted_____

437 #
438 # get filelist
439 #
440 if [ ! -f "$ALT_ROOT/boot/solaris/filelist.ramdisk" ] &&
441 [ ! -f "$ALT_ROOT/etc/boot/solaris/filelist.ramdisk" ]
442 then
443     print -u2 "Can't find filelist.ramdisk"
444     exit 1
445 fi
446 filelist=$(EXTRACT_FILELIST $EXTRACT_ARGS \
447 /boot/solaris/filelist.ramdisk \
448 /etc/boot/solaris/filelist.ramdisk \
449 2>/dev/null | sort -u)

451 #
452 # We use /tmp/ for scratch space now. This may be changed later if there
453 # is insufficient space in /tmp/.
454 #
455 rddir="/tmp/create_ramdisk.$$tmp"
456 new_rddir=
457 rm -rf "$rddir"
458 mkdir "$rddir" || fatal_error "Could not create temporary directory $rddir"

460 # Clean up upon exit.
461 trap 'cleanup' EXIT

463 list32="$rddir/filelist.32"
464 list64="$rddir/filelist.64"

466 touch $list32 $list64

468 #
469 # This loop creates the 32-bit and 64-bit lists of files. The 32-bit list
470 # is written to stdout, which is redirected at the end of the loop. The
471 # 64-bit list is appended with each write.
472 #
473 cd "/$ALT_ROOT"
474 find $filelist -print 2>/dev/null | while read path
475 do
476     if [ $SPLIT = no ] ; then
477         print "$path"

```

```

478     elif [ -d "$path" ]; then
479         if [ $format = ufs ]; then
480             size=`ls -lLd "$path" | /usr/xpg4/bin/awk '
480             size=`ls -lLd "$path" | awk '
481             {print ($5 % 1024) ? (int($5 / 1024) + 1) * 1024 : $
482             if [ `basename "$path"` != "amd64" ]; then
483                 (( dirsize32 += size ))
484             fi
485             (( dirsize64 += size ))
486         fi
487     else
488         case `LC_MESSAGES=C /usr/bin/file -m /dev/null "$path" 2>/dev/nu
489         *ELF\ 64-bit*)
490             print "$path" >> "$list64"
491             ;;
492         *ELF\ 32-bit*)
493             print "$path"
494             ;;
495         *)
496             # put in both lists
497             print "$path"
498             print "$path" >> "$list64"
499         esac
500     fi
501 done >"$list32"

503 if [ $format = ufs ]; then
504     # calculate image size
505     getsize

507     # check to see if there is sufficient space in tmpfs
508     #
509     tmp_free=`df -b /tmp | tail -1 | awk '{ printf ($2) }'`
510     (( tmp_free = tmp_free / 3 ))
511     if [ $SPLIT = yes ]; then
512         (( tmp_free = tmp_free / 2 ))
513     fi

515     if [ $total_size -gt $tmp_free ]; then
516         # assumes we have enough scratch space on $ALT_ROOT
517         new_rddir="/$ALT_ROOT/var/tmp/create_ramdisk.$$tmp"
518         rm -rf "$new_rddir"
519         mkdir "$new_rddir" || fatal_error \
520             "Could not create temporary directory $new_rddir"

522         # Save the file lists
523         mv "$list32" "$new_rddir"/
524         mv "$list64" "$new_rddir"/
525         list32="/$new_rddir/filelist.32"
526         list64="/$new_rddir/filelist.64"

528         # Remove the old $rddir and set the new value of rddir
529         rm -rf "$rddir"
530         rddir="$new_rddir"
531         new_rddir=
532     fi
533 fi

535 rdfile32="$rddir/rd.file.32"
536 rdfile64="$rddir/rd.file.64"
537 rdmnt32="$rddir/rd.mount.32"
538 rdmnt64="$rddir/rd.mount.64"
539 errlog32="$rddir/rd.errlog.32"
540 errlog64="$rddir/rd.errlog.64"
541 lofidev32=""
542 lofidev64=""

```

```

544 if [ $SPLIT = yes ]; then
545     #
546     # We can't run lofiadm commands in parallel, so we have to do
547     # them here.
548     #
549     if [ "$format" = "ufs" ]; then
550         mkfile ${size32}k "$rdfile32"
551         lofidev32=`lofiadm -a "$rdfile32"`
552         mkfile ${size64}k "$rdfile64"
553         lofidev64=`lofiadm -a "$rdfile64"`
554     fi
555     create_archive "32-bit" "$ALT_ROOT/$BOOT_ARCHIVE" $lofidev32 &
556     create_archive "64-bit" "$ALT_ROOT/$BOOT_ARCHIVE_64" $lofidev64
557     wait
558     if [ "$format" = "ufs" ]; then
559         lofiadm -d "$rdfile32"
560         lofiadm -d "$rdfile64"
561     fi
562 else
563     if [ "$format" = "ufs" ]; then
564         mkfile ${total_size}k "$rdfile32"
565         lofidev32=`lofiadm -a "$rdfile32"`
566     fi
567     create_archive "both" "$ALT_ROOT/$BOOT_ARCHIVE" $lofidev32
568     [ "$format" = "ufs" ] && lofiadm -d "$rdfile32"
569 fi
570 if [ $ERROR = 1 ]; then
571     cleanup
572     exit 1
573 fi

575 #
576 # For the diskless case, hardlink archive to /boot to make it
577 # visible via tftp. /boot is lofs mounted under /tftpboot/<hostname>.
578 # NOTE: this script must work on both client and server.
579 #
580 grep "[ ]/[ ]*"nfs[ ]" "$ALT_ROOT/etc/vfstab" > /dev/null
581 if [ $? = 0 ]; then
582     rm -f "$ALT_ROOT/boot/boot_archive" "$ALT_ROOT/boot/amd64/boot_archive"
583     ln "$ALT_ROOT/$BOOT_ARCHIVE" "$ALT_ROOT/boot/boot_archive"
584     if [ $SPLIT = yes ]; then
585         ln "$ALT_ROOT/$BOOT_ARCHIVE_64" \
586             "$ALT_ROOT/boot/amd64/boot_archive"
587     fi
588 fi
589 [ -n "$rddir" ] && rm -rf "$rddir"

```

new/usr/src/cmd/boot/scripts/root_archive.ksh

1

```
*****
7505 Sun May 26 18:22:36 2013
new/usr/src/cmd/boot/scripts/root_archive.ksh
XXX Remove awk(1)
*****
_____unchanged_portion_omitted_____

195 pack()
196 {
197     MR="$1"
198     [ -d "$UNPACKED_ROOT" ] || usage
199
200     # always compress if fiocompress exists
201     #
202     if [ -x /usr/sbin/fiocompress ] ; then
203         COMPRESS=true
204     fi
205
206     # Estimate image size and add %10 overhead for ufs stuff.
207     # Note, we can't use du here in case $UNPACKED_ROOT is on a filesystem,
208     # e.g. zfs, in which the disk usage is less than the sum of the file
209     # sizes. The awk code
210     # sizes. The awk code
211     #
212     # {t += ($7 % 1024) ? (int($7 / 1024) + 1) * 1024 : $7}
213     #
214     # below rounds up the size of a file/directory, in bytes, to the
215     # next multiple of 1024. This mimics the behavior of ufs especially
216     # with directories. This results in a total size that's slightly
217     # bigger than if du was called on a ufs directory.
218     #
219     # if the operation in turn is compressing the files the amount
220     # of typical shrinkage is used to come up with a useful archive
221     # size
222     size=$(find "$UNPACKED_ROOT" -ls | /usr/xpg4/bin/awk '
223     size=$(find "$UNPACKED_ROOT" -ls | awk '
224     {t += ($7 % 1024) ? (int($7 / 1024) + 1) * 1024 : $7}
225     END {print int(t * 1.10 / 1024)}')
226     if [ "$COMPRESS" = true ] ; then
227         size='echo $size | /usr/xpg4/bin/awk '{s = $1} END {print int(s
228         size='echo $size | awk '{s = $1} END {print int(s * 0.6)}''
229     fi
230
231     /usr/sbin/mkfile ${size}k "$TMR"
232
233     LOFIDEV='/usr/sbin/lofiadm -a "$TMR"'
234     if [ $? != 0 ] ; then
235         echo lofi plumb failed
236         exit 2
237     fi
238
239     RLOFIDEV='echo $LOFIDEV | sed s/lofi/rlofi/'
240     newfs $RLOFIDEV < /dev/null 2> /dev/null
241     mkdir -p $MNT
242     mount -o nologging $LOFIDEV $MNT
243     rmdir $MNT/lost+found
244
245     if [ -d "$UNPACKED_ROOT/kernel/drv/sparcv9" ] ; then
246         root_is_ramdisk $size
247     fi
248
249     (
250         cd "$UNPACKED_ROOT"
251         if [ "$COMPRESS" = true ] ; then
252             compress . $MNT
253         else
```

new/usr/src/cmd/boot/scripts/root_archive.ksh

2

```
251         find . -print | cpio -pdm $MNT 2> /dev/null
252     fi
253 )
254 lockfs -f $MNT
255 umount $MNT
256 rmdir $MNT
257
258 if [ -d "$UNPACKED_ROOT/kernel/drv/sparcv9" ] ; then
259     "$UNPACKED_ROOT/usr/sbin/installboot" \
260     "$UNPACKED_ROOT/platform/sun4u/lib/fs/ufs/bootblk" \
261     $RLOFIDEV
262 fi
263
264 lofiadm -d $LOFIDEV
265 LOFIDEV=
266
267 rm -f "$TMR.gz"
268
269 if [ -d "$UNPACKED_ROOT/kernel/drv/sparcv9" ] ; then
270     mv "$TMR" "$MR"
271 else
272     gzip -f "$TMR"
273     mv "$TMR.gz" "$MR"
274 fi
275
276 chmod a+r "$MR"
277 }
_____unchanged_portion_omitted_____
```

```

*****
4237 Sun May 26 18:22:37 2013
new/usr/src/cmd/boot/scripts/update_grub.ksh
XXX Remove awk(1)
*****
1 #!/bin/ksh -p
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 #
24 # Copyright 2008 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 #
28 PATH="/usr/bin:/usr/sbin:${PATH}"; export PATH
29 ALT_ROOT=
30 #
31 while getopts R: OPT 2>/dev/null
32 do
33     case $OPT in
34         R)      ALT_ROOT="$OPTARG"
35                ;;
36         ?)      echo "Usage: ${0##*/}: [-R <root>]"
37                ;;
38         esac
39 done
40 #
41 ARCH='uname -p'
42 #
43 is_pcfs_boot=yes
44 is_zfs_boot=no
45 #
46 check_pcfs_boot()
47 {
48     bootdev='grep -v "^#" "$ALT_ROOT"/etc/vfstab | grep pcfs \
49             | grep "[ ]/stubboot[ ]" | /usr/xpg4/bin/awk '{print $1}'
50             | grep "[ ]/stubboot[ ]" | /usr/xpg4/bin/awk '{print $1}'
51     if [ X"$bootdev" = "X" ]; then
52         is_pcfs_boot=no
53     fi
54 }
55 #
56 unchanged_portion_omitted
57 #
58 #
59 #
60 #
61 #
62 #
63 # Detect SVM root and return the list of raw devices under the mirror
64 #
65 get_rootdev_list()
66 {

```

```

77     if [ -f "$ALT_ROOT"/etc/lu/GRUB_slice ]; then
78         dev='grep '^PHYS_SLICE' "$ALT_ROOT"/etc/lu/GRUB_slice |
79             cut -d= -f2'
80         if [ "$is_zfs_boot" = "yes" ]; then
81             fstyp -a "$dev" | grep 'path: ' | grep -v phys_path: |
82                 cut -d'"' -f2 | sed 's+/dsk+/rdsk/+'
83         else
84             echo "$dev"
85         fi
86     fi
87     return
88     elif [ "$is_zfs_boot" = "yes" ]; then
89         rootpool='df -k ${ALT_ROOT:-/} | tail +2 | cut -d/ -f1'
90         rootdevlist='LC_ALL=C zpool iostat -v "$rootpool" | tail +5 |
91             egrep -v "mirror|spare|replacing" |
92             sed -n -e '/--/q' -e p | awk '{print $1}'
93     else
94         metadev='grep -v "^#" "$ALT_ROOT"/etc/vfstab | \
95             grep "[ ]/[ ]" | /usr/xpg4/bin/awk '{print $2}'
96             | /usr/xpg4/bin/awk '{print $2}'
97         if [[ $metadev = /dev/rdsk/* ]]; then
98             rootdevlist='basename "$metadev"'
99         elif [[ $metadev = /dev/md/rdsk/* ]]; then
100             metavol='basename "$metadev"'
101             rootdevlist='metastat -p $metavol | \
102                 grep -v "^$metavol[ ]" | /usr/xpg4/bin/awk '{print
103                 | /usr/xpg4/bin/awk '{print $4}'
104         fi
105     fi
106     for rootdev in $rootdevlist
107     do
108         echo /dev/rdsk/'basename $rootdev'
109     done
110 }
111 #
112 # multiboot: install grub on the boot slice
113 #
114 install_grub()
115 {
116     # Stage 2 blocks must remain untouched
117     STAGE1="$ALT_ROOT"/boot/grub/stage1
118     STAGE2="$ALT_ROOT"/boot/grub/stage2
119     #
120     # Note: /stubboot/boot/grub/stage2 must stay untouched.
121     #
122     mkdir -p "$ALT_ROOT"/stubboot/boot/grub
123     cp "$ALT_ROOT"/boot/grub/menu.lst "$ALT_ROOT"/stubboot/boot/grub
124     bootdev='grep -v "^#" "$ALT_ROOT"/etc/vfstab | grep pcfs | \
125             | /usr/xpg4/bin/awk '{print $1}'
126             | /usr/xpg4/bin/awk '{print $1}'
127     rpcfsdev='echo "$bootdev" | sed -e "s/dev\dsk/dev\rdsk/"
128     if [ X"$rpcfsdev" != X ]; then
129         print "Installing grub on $rpcfsdev"
130         "$ALT_ROOT"/sbin/installgrub $STAGE1 $STAGE2 $rpcfsdev
131     fi
132 }
133 #
134 #
135 #
136 #
137 #
138 #
139 #
140 #
141 #
142 #
143 #
144 #
145 #
146 #
147 #
148 #
149 #
150 #
151 #
152 #
153 #
154 #
155 #
156 #
157 #
158 #
159 #
160 #
161 #
162 #
163 #
164 #
165 #
166 #
167 #
168 #
169 #
170 #
171 #
172 #
173 #
174 #
175 #
176 #
177 #
178 #
179 #
180 #
181 #
182 #
183 #
184 #
185 #
186 #
187 #
188 #
189 #
190 #
191 #
192 #
193 #
194 #
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #
221 #
222 #
223 #
224 #
225 #
226 #
227 #
228 #
229 #
230 #
231 #
232 #
233 #
234 #
235 #
236 #
237 #
238 #
239 #
240 #
241 #
242 #
243 #
244 #
245 #
246 #
247 #
248 #
249 #
250 #
251 #
252 #
253 #
254 #
255 #
256 #
257 #
258 #
259 #
260 #
261 #
262 #
263 #
264 #
265 #
266 #
267 #
268 #
269 #
270 #
271 #
272 #
273 #
274 #
275 #
276 #
277 #
278 #
279 #
280 #
281 #
282 #
283 #
284 #
285 #
286 #
287 #
288 #
289 #
290 #
291 #
292 #
293 #
294 #
295 #
296 #
297 #
298 #
299 #
300 #
301 #
302 #
303 #
304 #
305 #
306 #
307 #
308 #
309 #
310 #
311 #
312 #
313 #
314 #
315 #
316 #
317 #
318 #
319 #
320 #
321 #
322 #
323 #
324 #
325 #
326 #
327 #
328 #
329 #
330 #
331 #
332 #
333 #
334 #
335 #
336 #
337 #
338 #
339 #
340 #
341 #
342 #
343 #
344 #
345 #
346 #
347 #
348 #
349 #
350 #
351 #
352 #
353 #
354 #
355 #
356 #
357 #
358 #
359 #
360 #
361 #
362 #
363 #
364 #
365 #
366 #
367 #
368 #
369 #
370 #
371 #
372 #
373 #
374 #
375 #
376 #
377 #
378 #
379 #
380 #
381 #
382 #
383 #
384 #
385 #
386 #
387 #
388 #
389 #
390 #
391 #
392 #
393 #
394 #
395 #
396 #
397 #
398 #
399 #
400 #
401 #
402 #
403 #
404 #
405 #
406 #
407 #
408 #
409 #
410 #
411 #
412 #
413 #
414 #
415 #
416 #
417 #
418 #
419 #
420 #
421 #
422 #
423 #
424 #
425 #
426 #
427 #
428 #
429 #
430 #
431 #
432 #
433 #
434 #
435 #
436 #
437 #
438 #
439 #
440 #
441 #
442 #
443 #
444 #
445 #
446 #
447 #
448 #
449 #
450 #
451 #
452 #
453 #
454 #
455 #
456 #
457 #
458 #
459 #
460 #
461 #
462 #
463 #
464 #
465 #
466 #
467 #
468 #
469 #
470 #
471 #
472 #
473 #
474 #
475 #
476 #
477 #
478 #
479 #
480 #
481 #
482 #
483 #
484 #
485 #
486 #
487 #
488 #
489 #
490 #
491 #
492 #
493 #
494 #
495 #
496 #
497 #
498 #
499 #
500 #
501 #
502 #
503 #
504 #
505 #
506 #
507 #
508 #
509 #
510 #
511 #
512 #
513 #
514 #
515 #
516 #
517 #
518 #
519 #
520 #
521 #
522 #
523 #
524 #
525 #
526 #
527 #
528 #
529 #
530 #
531 #
532 #
533 #
534 #
535 #
536 #
537 #
538 #
539 #
540 #
541 #
542 #
543 #
544 #
545 #
546 #
547 #
548 #
549 #
550 #
551 #
552 #
553 #
554 #
555 #
556 #
557 #
558 #
559 #
560 #
561 #
562 #
563 #
564 #
565 #
566 #
567 #
568 #
569 #
570 #
571 #
572 #
573 #
574 #
575 #
576 #
577 #
578 #
579 #
580 #
581 #
582 #
583 #
584 #
585 #
586 #
587 #
588 #
589 #
590 #
591 #
592 #
593 #
594 #
595 #
596 #
597 #
598 #
599 #
600 #
601 #
602 #
603 #
604 #
605 #
606 #
607 #
608 #
609 #
610 #
611 #
612 #
613 #
614 #
615 #
616 #
617 #
618 #
619 #
620 #
621 #
622 #
623 #
624 #
625 #
626 #
627 #
628 #
629 #
630 #
631 #
632 #
633 #
634 #
635 #
636 #
637 #
638 #
639 #
640 #
641 #
642 #
643 #
644 #
645 #
646 #
647 #
648 #
649 #
650 #
651 #
652 #
653 #
654 #
655 #
656 #
657 #
658 #
659 #
660 #
661 #
662 #
663 #
664 #
665 #
666 #
667 #
668 #
669 #
670 #
671 #
672 #
673 #
674 #
675 #
676 #
677 #
678 #
679 #
680 #
681 #
682 #
683 #
684 #
685 #
686 #
687 #
688 #
689 #
690 #
691 #
692 #
693 #
694 #
695 #
696 #
697 #
698 #
699 #
700 #
701 #
702 #
703 #
704 #
705 #
706 #
707 #
708 #
709 #
710 #
711 #
712 #
713 #
714 #
715 #
716 #
717 #
718 #
719 #
720 #
721 #
722 #
723 #
724 #
725 #
726 #
727 #
728 #
729 #
730 #
731 #
732 #
733 #
734 #
735 #
736 #
737 #
738 #
739 #
740 #
741 #
742 #
743 #
744 #
745 #
746 #
747 #
748 #
749 #
750 #
751 #
752 #
753 #
754 #
755 #
756 #
757 #
758 #
759 #
760 #
761 #
762 #
763 #
764 #
765 #
766 #
767 #
768 #
769 #
770 #
771 #
772 #
773 #
774 #
775 #
776 #
777 #
778 #
779 #
780 #
781 #
782 #
783 #
784 #
785 #
786 #
787 #
788 #
789 #
790 #
791 #
792 #
793 #
794 #
795 #
796 #
797 #
798 #
799 #
800 #
801 #
802 #
803 #
804 #
805 #
806 #
807 #
808 #
809 #
810 #
811 #
812 #
813 #
814 #
815 #
816 #
817 #
818 #
819 #
820 #
821 #
822 #
823 #
824 #
825 #
826 #
827 #
828 #
829 #
830 #
831 #
832 #
833 #
834 #
835 #
836 #
837 #
838 #
839 #
840 #
841 #
842 #
843 #
844 #
845 #
846 #
847 #
848 #
849 #
850 #
851 #
852 #
853 #
854 #
855 #
856 #
857 #
858 #
859 #
860 #
861 #
862 #
863 #
864 #
865 #
866 #
867 #
868 #
869 #
870 #
871 #
872 #
873 #
874 #
875 #
876 #
877 #
878 #
879 #
880 #
881 #
882 #
883 #
884 #
885 #
886 #
887 #
888 #
889 #
890 #
891 #
892 #
893 #
894 #
895 #
896 #
897 #
898 #
899 #
900 #
901 #
902 #
903 #
904 #
905 #
906 #
907 #
908 #
909 #
910 #
911 #
912 #
913 #
914 #
915 #
916 #
917 #
918 #
919 #
920 #
921 #
922 #
923 #
924 #
925 #
926 #
927 #
928 #
929 #
930 #
931 #
932 #
933 #
934 #
935 #
936 #
937 #
938 #
939 #
940 #
941 #
942 #
943 #
944 #
945 #
946 #
947 #
948 #
949 #
950 #
951 #
952 #
953 #
954 #
955 #
956 #
957 #
958 #
959 #
960 #
961 #
962 #
963 #
964 #
965 #
966 #
967 #
968 #
969 #
970 #
971 #
972 #
973 #
974 #
975 #
976 #
977 #
978 #
979 #
980 #
981 #
982 #
983 #
984 #
985 #
986 #
987 #
988 #
989 #
990 #
991 #
992 #
993 #
994 #
995 #
996 #
997 #
998 #
999 #
1000 #

```

```
140     do
141         if [ X"$rpcfsdev" != X ]; then
142             echo "create GRUB menu in "$ALT_ROOT"/stubboot"
143             "$ALT_ROOT"/sbin/bootadm update-menu $zfsarg \
144                 -R "$ALT_ROOT"/stubboot -o $rootdev,"$ALT_ROOT"
145         else
146             echo "Creating GRUB menu in ${ALT_ROOT:-/}"
147             "$ALT_ROOT"/sbin/bootadm update-menu -R ${ALT_ROOT:-/} \
148                 $zfsarg -o $rootdev
149         fi
150         print "Installing grub on $rootdev"
151         "$ALT_ROOT"/sbin/installgrub $STAGE1 $STAGE2 $rootdev
152     done
153 }
_____unchanged_portion_omitted_____
```

new/usr/src/cmd/cmd-inet/etc/init.d/pppd

1

```
*****
2662 Sun May 26 18:22:37 2013
new/usr/src/cmd/cmd-inet/etc/init.d/pppd
XXX Remove awk(1)
*****
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 PATH=/sbin:/usr/bin:/usr/sbin: export PATH
27 PPPDIR=/etc/ppp: export PPPDIR
28
30 case "$1" in
31 'start')
32     if [ ! -x /usr/bin/pppd -o ! -c /dev/sppp ]; then
33         echo "$0: Solaris PPP has not been correctly installed on"
34         echo "$0: this system. Required files are missing."
35         exit 1
36     fi
37     if [ -f $PPPDIR/ifconfig ]; then
38         . $PPPDIR/ifconfig
39     fi
40     if [ -f $PPPDIR/demand ]; then
41         . $PPPDIR/demand
42     fi
43     if [ -f $PPPDIR/pppoe.if ] && [ -x /usr/sbin/sppptun ]; then
44         sed -e 's/^#.*//;s/([^\]\\)#.*\1/is/[ ]*$/;s/^[ ]*$/ \
45             $PPPDIR/pppoe.if | \
46             while read intf saps sapd; do
47                 if [ "$intf" ]; then
48                     [ -z "$saps" ] || saps="-s $saps"
49                     /usr/sbin/sppptun plumb $saps pppoe $intf
50                     [ -z "$sapd" ] || sapd="-s $sapd"
51                     /usr/sbin/sppptun plumb $sapd pppoe $intf
52                 fi
53             done
54     fi
55     if [ -f $PPPDIR/pppoe ] && [ -x /usr/lib/inet/pppoed ]; then
56         /usr/lib/inet/pppoed >/dev/null
57     fi
58     ;;
59
60 'stop')
61     /usr/bin/pkill -z '/sbin/zonename' -x pppd && sleep 1
```

new/usr/src/cmd/cmd-inet/etc/init.d/pppd

2

```
62     /usr/bin/pkill -z '/sbin/zonename' -x pppoe
63
64     # Use ifconfig to make the interfaces down just in case
65     if [ -f $PPPDIR/ifconfig ]; then
66         /usr/xpg4/bin/awk '/ifconfig[ ]*sppp/ { \
67             awk '/ifconfig[ ]*sppp/ { \
68                 system("ifconfig " $2 " down"); \
69                 system("ifconfig " $2 " unplumb"); \
70                 next; \
71             } \
72             /ifconfig/ { \
73                 $3 = "removeif"; \
74                 NF = 4; \
75                 system($0); \
76             }' < $PPPDIR/ifconfig
77     fi
78
79     if [ -f $PPPDIR/pppoe.if ] && [ -x /usr/sbin/sppptun ]; then
80         sed -e 's/^#.*//;s/([^\]\\)#.*\1/is/[ ]*$/;s/^[ ]*$/ \
81             $PPPDIR/pppoe.if | \
82             while read intf rest; do
83                 if [ "$intf" ]; then
84                     /usr/sbin/sppptun unplumb ${intf}:pppoe
85                     /usr/sbin/sppptun unplumb ${intf}:pppoed
86                 fi
87             done
88     fi
89
90 *)
91     echo "Usage: $0 { start | stop }"
92     exit 1
93     ;;
94 esac
95 exit 0
```

```

*****
2815 Sun May 26 18:22:38 2013
new/usr/src/cmd/cmd-inet/usr.sbin/routedm/svc-legacy-routing
XXX Remove awk(1)
*****
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # ident "%Z%M% %I% %E% SMI"

27 # This script is the shared method script for the legacy ipv4/ipv6
28 # routing services.

30 . /lib/svc/share/smf_include.sh

32 daemon_prog='\usr/sbin/svccfg -s $SMF_FMRI listprop routedm/daemon | \
33 /usr/xpg4/bin/awk '{ for (i = 3; i <= NF; i++) printf $i " " }' | \
34 /usr/xpg4/bin/awk '{ sub(/^\"/, ""); sub(/\\"[ \t]*$/, ""); print }' | \
35 /usr/bin/nawk '{ for (i = 3; i <= NF; i++) printf $i " " }' | \
36 /usr/bin/nawk '{ sub(/^\"/, ""); sub(/\\"[ \t]*$/, ""); print }'
35 daemon_args='\usr/sbin/svccfg -s $SMF_FMRI listprop routedm/daemon-args | \
36 /usr/xpg4/bin/awk '{ for (i = 3; i <= NF; i++) printf $i " " }' | \
37 /usr/xpg4/bin/awk '{ sub(/^\"/, ""); sub(/\\"[ \t]*$/, ""); print }' | \
38 /usr/bin/nawk '{ for (i = 3; i <= NF; i++) printf $i " " }' | \
39 /usr/bin/nawk '{ sub(/^\"/, ""); sub(/\\"[ \t]*$/, ""); print }'
38 daemon_stop='\usr/sbin/svccfg -s $SMF_FMRI listprop routedm/daemon-stop-cmd | \
39 /usr/xpg4/bin/awk '{ for (i = 3; i <= NF; i++) printf $i " " }' | \
40 /usr/xpg4/bin/awk '{ sub(/^\"/, ""); sub(/\\"[ \t]*$/, ""); print }' | \
41 /usr/bin/nawk '{ for (i = 3; i <= NF; i++) printf $i " " }' | \
42 /usr/bin/nawk '{ sub(/^\"/, ""); sub(/\\"[ \t]*$/, ""); print }'

42 method="$1"
43 proto="$2"

45 case "$method" in
46 'start' )
47     # No legacy daemon specified.
48     if [ -z "$daemon_prog" ]; then
49         echo "${proto}-routing-daemon not specified by routedm."
50         exit $SMF_EXIT_ERR_CONFIG
51     fi
52     # No legacy stop command specified.
53     if [ -z "$daemon_stop" ]; then
54         echo "${proto}-routing-stop-cmd not specified by routedm."

```

```

55     exit $SMF_EXIT_ERR_CONFIG
56 fi
57 smf_configure_ip || exit $SMF_EXIT_OK

59 # Run daemon - fail if it does not successfully daemonize.
60 eval "$daemon_prog $daemon_args"
61 if [ "$?" != "0" ]; then
62     echo "Error: $daemon $daemon_args failed to daemonize."
63     exit $SMF_EXIT_ERR_FATAL
64 fi
65 # Create pidfile.
66 daemon_name='\usr/bin/basename $daemon_prog'
67 if smf_is_globalzone; then
68     /usr/bin/pgrep -P 1 -z 'smf_zonename' -f $daemon_prog > \
69     /var/tmp/${daemon_name}.pid
70 else
71     /usr/bin/pgrep -z 'smf_zonename' -f $daemon_prog > \
72     /var/tmp/${daemon_name}.pid
73 fi
74 ;;
75 'stop' )
76 smf_configure_ip || exit $SMF_EXIT_OK

78 # Stop daemon - ignore result.
79 if [ -n "$daemon_stop" ]; then
80     eval "$daemon_stop"
81 fi
82 ;;
83 '*' )
84 echo "Usage: $0 { start | stop }"
85 exit $SMF_EXIT_ERR_FATAL
86 ;;
87 esac

89 exit "$SMF_EXIT_OK"

```


new/usr/src/cmd/dtrace/test/cmd/scripts/dtfailures.ksh

1

1306 Sun May 26 18:22:38 2013

new/usr/src/cmd/dtrace/test/cmd/scripts/dtfailures.ksh

XXX Remove awk(1)

```
1 #!/usr/bin/ksh -p
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 #
24 # Copyright (c) 2011, Joyent, Inc. All rights reserved.
25 #
26 #
27 let failure=0
28 #
29 printf "%-3s %-10s %-31s %s\n" "#" "KIND" "TEST" "DETAILS"
30 #
31 while [[ -d failure.$failure ]]; do
32   dir=failure.$failure
33   tst='cat $dir/README | head -1 | /usr/xpg4/bin/awk '{ print $2 }''
34   tst='cat $dir/README | head -1 | awk '{ print $2 }''
35   kind='basename $(dirname $tst)'
36   name='basename $tst'
37   cols=$(expr `tput cols` - 47)
38   details='tail -1 $dir/*.err | cut -c1-$cols'
39   printf "%-3d %-10s %-31s " $failure $kind $name
40   echo $details
41   let failure=failure+1
42 done
```

new/usr/src/cmd/dtrace/test/tst/common/funcs/tst.badfreopen.ksh

1

```
*****
1981 Sun May 26 18:22:38 2013
new/usr/src/cmd/dtrace/test/tst/common/funcs/tst.badfreopen.ksh
XXX Remove awk(1)
*****
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 2006 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #ident "%Z%M% %I% %E% SMI"

27 script()
28 {
29     $dtrace -wq -o $tmpfile -s /dev/stdin 2> $errfile <<EOF
30     BEGIN
31     {
32         /*
33          * All of these should fail...
34          */
35         freopen("..");
36         freopen("%s", ".");
37         freopen("%c%c", '.', '.');
38         freopen("%c", '.');

40         /*
41          * ...so stdout should still be open here.
42          */
43         printf("%d", ++i);

45         freopen("%s%s", ".", ".");
46         freopen("%s%s", ".", ".");

48         printf("%d", ++i);
49     }

51     BEGIN
52     /i == 2/
53     {
54         /*
55          * ...and here.
56          */
57         printf("%d\n", ++i);
58         exit(0);
59     }

```

new/usr/src/cmd/dtrace/test/tst/common/funcs/tst.badfreopen.ksh

2

```
61     BEGIN
62     {
63         exit(1);
64     }
65 EOF
66 }

68 if [ $# != 1 ]; then
69     echo expected one argument: '<dtrace-path>'
70     exit 2
71 fi

73 dtrace=$1
74 tmpfile=/tmp/tst.badfreopen.$$
75 errfile=/tmp/tst.badfreopen.$$stderr

77 script
78 status=$?

80 if [ "$status" -eq 0 ]; then
81     i='cat $tmpfile'

83     if [[ $i != "123" ]]; then
84         echo "$0: unexpected contents in $tmpfile: " \
85             "expected 123, found $i"
86         status=100
87     fi
88

89     i='wc -l $errfile | /usr/xpg4/bin/awk '{ print $1 }''
90     i='wc -l $errfile | awk '{ print $1 }''

91     if [ "$i" -lt 6 ]; then
92         echo "$0: expected at least 6 lines of stderr, found $i lines"
93         status=101
94     fi
95 else
96     cat $errfile > /dev/fd/2
97 fi

99 rm $tmpfile $errfile

101 exit $status

```

new/usr/src/cmd/dtrace/test/tst/i86xpv/xdt/tst.schedargs.ksh

1

2269 Sun May 26 18:22:39 2013

new/usr/src/cmd/dtrace/test/tst/i86xpv/xdt/tst.schedargs.ksh

XXX Remove awk(1)

unchanged_portion_omitted_

```
62 validate()
63 {
64     /usr/xpg4/bin/awk '
64     /usr/bin/nawk '
65     BEGIN {
66         while (("usr/sbin/xm vcpu-list" | getline)) {
67             if ($1 != "Name") {
68                 domid = $2
69                 vcpu = $3
71                 vcpumap[domid, vcpu] = 1
73                 split($7, affinity, ",")
74                 for (i in affinity) {
75                     if (split(affinity[i], p, "-") > 1) {
76                         for (pcpu = p[1]; pcpu <= p[2];\
77                             pcpu++) {
78                             cpumap[domid, vcpu,
79                                 pcpu] = 1
80                         }
81                     } else {
82                         cpumap[domid, vcpu,
83                             affinity[i]] = 1
84                     }
85                 }
86             }
87         }
88     }
90     /^$/ { next }
92     /wake/ {
93         if (vcpumap[$1, $2]) {
94             next
95         } else {
96             print "error: " $0
97             exit 1
98         }
99     }
101     {
102         if (cpumap[$1, $2, "any"] || cpumap[$1, $2, $3]) {
103             next
104         } else {
105             print "error: " $0
106             exit 1
107         }
108     }
109     , $outf
110 }
```

unchanged_portion_omitted_

new/usr/src/cmd/fs.d/nfs/nfsfind/nfsfind.sh

1

1678 Sun May 26 18:22:39 2013

new/usr/src/cmd/fs.d/nfs/nfsfind/nfsfind.sh

XXX Remove awk(1)

```
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License, Version 1.0 only
7 # (the "License"). You may not use this file except in compliance
8 # with the License.
9 #
10 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
11 # or http://www.opensolaris.org/os/licensing.
12 # See the License for the specific language governing permissions
13 # and limitations under the License.
14 #
15 # When distributing Covered Code, include this CDDL HEADER in each
16 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
17 # If applicable, add the following below this CDDL HEADER, with the
18 # fields enclosed by brackets "[]" replaced with your own identifying
19 # information: Portions Copyright [yyyy] [name of copyright owner]
20 #
21 # CDDL HEADER END
22 #
23 # Copyright (c) 1993 by Sun Microsystems, Inc.

25 #ident "%Z%M% %I% %E% SMI"
26 #
27 # Check shared NFS filesystems for .nfs* files that
28 # are more than a week old.
29 #
30 # These files are created by NFS clients when an open file
31 # is removed. To preserve some semblance of Unix semantics
32 # the client renames the file to a unique name so that the
33 # file appears to have been removed from the directory, but
34 # is still usable by the process that has the file open.

36 if [ ! -s /etc/dfs/sharetab ]; then exit ; fi

38 # Get all NFS filesystems exported with read-write permission.

40 DIRS='/usr/xpg4/bin/awk '($3 != "nfs") { next }
40 DIRS='/usr/bin/nawk '($3 != "nfs") { next }
41 ($4 ~ /^rw$|^rw|^rw|^rw|^rw|^rw$/) { print $1; next }
42 ($4 !~ /^ro$|^ro|^ro|^ro|^ro$/) { print $1 }' /etc/dfs/sharetab'

44 for dir in $DIRS
45 do
46     find $dir -type f -name .nfs\* -mtime +7 -mount -exec rm -f {} \;
47 done
```

new/usr/src/cmd/getmajor/getmajor.sh

1

```
*****
1222 Sun May 26 18:22:40 2013
new/usr/src/cmd/getmajor/getmajor.sh
XXX Remove awk(1)
*****
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright 2006 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #ident "%Z%M% %I% %E% SMI"
27 if [ $# -ne 1 ]
28 then
29     echo "Usage: `basename $0` modname" >&2
30     exit 2
31 fi
32 #
33 exec /usr/xpg4/bin/awk -v a="$1" -v del='#' '$1 !~ /^#|^$/ && $1 == a { \
34 exec awk -v a="$1" -v del='#' '$1 !~ /^#|^$/ && $1 == a { \
34     num = split($2, maj, del); \
35     if (num > 1) { print maj[1] } else { print $2 } \
36     found = 1; exit 0 } \
37     END {if (!found) exit 1}' < /etc/name_to_major
```

```
new/usr/src/cmd/ldap/ns_ldap/idsconfig.sh
```

1

```
*****
```

```
171053 Sun May 26 18:22:40 2013
```

```
new/usr/src/cmd/ldap/ns_ldap/idsconfig.sh
```

```
XXX Remove nawk(1)
```

```
*****
```

```
_____unchanged_portion_omitted_____
```

```
961 #
962 # init(): initializes variables and options
963 #
964 init()
965 {
966     # General variables.
967     PROG='basename $0' # Program name
968     PID=$$ # Program ID
969     VERB='> /dev/null 2>&1' # NULL or "> /dev/null"
970     ECHO="/bin/echo" # print message on screen
971     EVAL="eval" # eval or echo
972     EGREP="/usr/bin/egrep"
973     GREP="/usr/bin/grep"
974     DEBUG=0 # Set Debug OFF
975     BACKUP=no_ldap # backup suffix
976     HOST="" # NULL or <hostname>
977     AWK="/usr/xpg4/bin/awk"
978     NAWK="/usr/bin/nawk"
979     RM="/usr/bin/rm"
980     WC="/usr/bin/wc"
981     CAT="/usr/bin/cat"
982     SED="/usr/bin/sed"
983     MV="/usr/bin/mv"
984
985     DOM="" # Set to NULL
986     # If DNS domain (resolv.conf) exists use that, otherwise use domainname.
987     if [ -f /etc/resolv.conf ]; then
988         DOM='/usr/xpg4/bin/grep -i -E '^domain|^search' /etc/resolv.conf \
989         | awk '{ print $2 }' | tail -1'
990     fi
991
992     # If for any reason the DOM did not get set (error'd resolv.conf) set
993     # DOM to the domainname command's output.
994     if [ "$DOM" = "" ]; then
995         DOM='domainname' # domain from domainname command.
996     fi
997
998     STEP=1
999     INTERACTIVE=1 # 0 = on, 1 = off (For input file mode)
1000     DEL_OLD_PROFILE=0 # 0 (default), 1 = delete old profile.
1001
1002     # idsconfig specific variables.
1003     INPUT_FILE=""
1004     OUTPUT_FILE=""
1005     LDAP_ENABLE_SHADOW_UPDATE="FALSE"
1006     NEED_PROXY=0 # 0 = No Proxy, 1 = Create Proxy.
1007     NEED_ADMIN=0 # 0 = No Admin, 1 = Create Admin.
1008     NEED_HOSTACL=0 # 0 = No Host ACL, 1 = Create Host ACL.
1009     EXISTING_PROFILE=0
1010     LDAP_PROXYAGENT=""
1011     LDAP_ADMINDN=""
1012     LDAP_SUFFIX=""
1013     LDAP_DOMAIN=$DOM # domainname on Server (default value)
1014     GEN_CMD=""
1015     PROXY_ACI_NAME="LDAP_Naming_Services_proxy_password_read"
1016
1017     # LDAP COMMANDS
1018     LDAPSEARCH="/bin/ldapsearch -r"
```

```
new/usr/src/cmd/ldap/ns_ldap/idsconfig.sh
```

2

```
1018 LDAPMODIFY=/bin/ldapmodify
1019 LDAPADD=/bin/ldapadd
1020 LDAPDELETE=/bin/ldapdelete
1021 LDAP_GEN_PROFILE=/usr/sbin/ldap_gen_profile
1022
1023 # IDS specific information
1024 IDS_SERVER=""
1025 IDS_PORT=389
1026 NEED_TIME=0
1027 NEED_SIZE=0
1028 NEED_SRVAUTH_PAM=0
1029 NEED_SRVAUTH_KEY=0
1030 NEED_SRVAUTH_CMD=0
1031 IDS_TIMELIMIT=""
1032 IDS_SIZELIMIT=""
1033
1034 # LDAP PROFILE related defaults
1035 LDAP_ROOTDN="cn=Directory Manager" # Provide common default.
1036 LDAP_ROOTPWD="" # NULL passwd as default (i.e. invalid)
1037 LDAP_PROFILE_NAME="default"
1038 LDAP_BASEDN=""
1039 LDAP_SERVER_LIST=""
1040 LDAP_AUTHMETHOD=""
1041 LDAP_FOLLOWREF="FALSE"
1042 NEED_CRYPT=""
1043 LDAP_SEARCH_SCOPE="one"
1044 LDAP_SRV_AUTHMETHOD_PAM=""
1045 LDAP_SRV_AUTHMETHOD_KEY=""
1046 LDAP_SRV_AUTHMETHOD_CMD=""
1047 LDAP_SEARCH_TIME_LIMIT=30
1048 LDAP_PREF_SRVLIST=""
1049 LDAP_PROFILE_TTL=43200
1050 LDAP_CRED_LEVEL="proxy"
1051 LDAP_BIND_LIMIT=10
1052
1053 # Prevent new files from being read by group or others.
1054 umask 077
1055
1056 # Service Search Descriptors
1057 LDAP_SERV_SRCH_DES=""
1058
1059 # Set and create TMPDIR.
1060 TMPDIR="/tmp/idsconfig.${PID}"
1061 if mkdir -m 700 ${TMPDIR}
1062 then
1063     # Cleanup on exit.
1064     trap 'rm -rf ${TMPDIR}; /usr/bin/stty echo; exit' 1 2 3 6 15
1065 else
1066     echo "ERROR: unable to create a safe temporary directory."
1067     exit 1
1068 fi
1069 LDAP_ROOTPWF=${TMPDIR}/rootPWD
1070
1071 # Set the SSD file name after setting TMPDIR.
1072 SSD_FILE=${TMPDIR}/ssd_list
1073
1074 # GSSAPI setup
1075 GSSAPI_ENABLE=0
1076 LDAP_KRB_REALM=""
1077 SCHEMA_UPDATED=0
1078
1079 export DEBUG VERB ECHO EVAL EGREP GREP STEP TMPDIR
1080 export IDS_SERVER IDS_PORT LDAP_ROOTDN LDAP_ROOTPWD LDAP_SERVER_LIST
1081 export LDAP_BASEDN LDAP_ROOTPWF
1082 export LDAP_DOMAIN LDAP_SUFFIX LDAP_PROXYAGENT LDAP_PROXYAGENT_CRED
1083 export NEED_PROXY
```

```

1084 export LDAP_ENABLE_SHADOW_UPDATE LDAP_ADMINDN LDAP_ADMIN_CRED
1085 export NEED_ADMIN NEED_HOSTACL EXISTING_PROFILE
1086 export LDAP_PROFILE_NAME LDAP_BASEDN LDAP_SERVER_LIST
1087 export LDAP_AUTHMETHOD LDAP_FOLLOWREF LDAP_SEARCH_SCOPE LDAP_SEARCH_TIME_LIM
1088 export LDAP_PREF_SRVLIST LDAP_PROFILE_TTL LDAP_CRED_LEVEL LDAP_BIND_LIMIT
1089 export NEED_SRVAUTH_PAM NEED_SRVAUTH_KEY NEED_SRVAUTH_CMD
1090 export LDAP_SRV_AUTHMETHOD_PAM LDAP_SRV_AUTHMETHOD_KEY LDAP_SRV_AUTHMETHOD_C
1091 export LDAP_SERV_SRCH_DES SSD_FILE
1092 export GEN_CMD GSSAPI_ENABLE LDAP_KRB_REALM SCHEMA_UPDATED
1093 }

```

unchanged portion omitted

```
1424 get_krb_realm() {
```

```

1426 # To upper cases
1427 LDAP_KRB_REALM=${ECHO} ${LDAP_DOMAIN} | ${AWK} '{ print toupper($0) }'
1428 LDAP_KRB_REALM=${ECHO} ${LDAP_DOMAIN} | ${NAWK} '{ print toupper($0) }'
1429 get_ans_req "Enter Kerberos Realm:" "$LDAP_KRB_REALM"
1429 # To upper cases
1430 LDAP_KRB_REALM=${ECHO} ${ANS} | ${AWK} '{ print toupper($0) }'
1431 LDAP_KRB_REALM=${ECHO} ${ANS} | ${NAWK} '{ print toupper($0) }'
1431 }

```

unchanged portion omitted

```
5048 # Delete Proxy Agent read permission for password.
```

```
5049 delete_proxy_read_pw()
```

```
5050 {
5051 [ $DEBUG -eq 1 ] && ${ECHO} "In delete_proxy_read_pw()"
```

```

5053 # Search for ACI_NAME
5054 eval "${LDAPSEARCH} ${LDAP_ARGS} -b \"${LDAP_BASEDN}\" -s base objectclass=*
5055 ${GREP} "${PROXY_ACI_NAME}" ${TMPDIR}/chk_proxyread_aci | \
5056 ${SED} -e 's/aci=/' > ${TMPDIR}/grep_proxyread_aci 2>&1
5057 if [ $? -ne 0 ]; then
5058   ${ECHO} "Proxy ACI ${PROXY_ACI_NAME} does not exist for ${LDAP_BASEDN}."
5059   return 0
5060 fi

```

```

5062 # We need to remove proxy agent's read access to user passwords,
5063 # but We do not know the value of the ${LDAP_PROXYAGENT} here, so
5064 # 1. if only one match found, delete it
5065 # 2. if more than one matches found, ask the user which one to delete
5066 HOWMANY=${WC} -l ${TMPDIR}/grep_proxyread_aci | ${AWK} '{print $1}'
5067 HOWMANY=${WC} -l ${TMPDIR}/grep_proxyread_aci | ${NAWK} '{print $1}'
5067 if [ $HOWMANY -eq 0 ]; then
5068   ${ECHO} "Proxy ACI ${PROXY_ACI_NAME} does not exist for ${LDAP_BASEDN}."
5069   return 0
5070 fi
5071 if [ $HOWMANY -eq 1 ];then
5072   proxy_aci=${CAT} ${TMPDIR}/grep_proxyread_aci
5073 else
5074   ${CAT} << EOF

```

```

5076 Proxy agent is not allowed to read user passwords when shadow
5077 update is enabled. There are more than one proxy agents found.
5078 Please select the currently proxy agent being used, so that
5079 idsconfig can remove its read access to user passwords.

```

```
5081 The proxy agents are:
```

```

5083 EOF
5084 # generate the proxy agent list
5085 ${SED} -e "s/.*ldap://\./.*ldap://\./" \
5086 ${TMPDIR}/grep_proxyread_aci | ${SED} -e "s/\";)" > \
5087 ${TMPDIR}/proxy_agent_list

```

```

5089 # print the proxy agent list
5090 ${AWK} '{print NR ": " $0}' ${TMPDIR}/proxy_agent_list
5090 ${NAWK} '{print NR ": " $0}' ${TMPDIR}/proxy_agent_list

```

```

5092 # ask the user to pick one
5093 _MENU_PROMPT="Select the proxy agent (1-$HOWMANY): "
5094 get_menu_choice "${_MENU_PROMPT}" "0" "$HOWMANY"
5095 _CH=${MN}_CH
5096 proxy_aci=${SED} -n "$_CH p" ${TMPDIR}/grep_proxyread_aci
5097 fi

```

```
5099 # Create the tmp file to delete the ACI.
```

```

5100 ( cat <<EOF
5101 dn: ${LDAP_BASEDN}
5102 changetype: modify
5103 delete: aci
5104 aci: ${proxy_aci}
5105 EOF
5106 ) > ${TMPDIR}/proxy_delete

```

```

5108 # Delete the ACI
5109 ${EVAL} "${LDAPMODIFY} ${LDAP_ARGS} -f ${TMPDIR}/proxy_delete ${VERB}"
5110 if [ $? -ne 0 ]; then
5111   ${ECHO} " ERROR: Remove of ${PROXY_ACI_NAME} ACI failed!"
5112   cat ${TMPDIR}/proxy_delete
5113   cleanup
5114   exit 1
5115 fi

```

```

5117 # Display message that ACI is updated.
5118 MSG="Removed ${PROXY_ACI_NAME} ACI for proxyagent read permission for passwo
5119 ${ECHO} " "
5120 ${ECHO} " ACI REMOVED: $MSG"
5121 ${ECHO} " The ACI removed is $proxy_aci"
5122 ${ECHO} " "
5123 }

```

unchanged portion omitted

new/usr/src/cmd/lvm/etc/svm.cleanup.sh

1

```
*****
7098 Sun May 26 18:22:41 2013
new/usr/src/cmd/lvm/etc/svm.cleanup.sh
XXX Remove nawk(1)
*****
1 #!/usr/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License, Version 1.0 only
7 # (the "License"). You may not use this file except in compliance
8 # with the License.
9 #
10 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
11 # or http://www.opensolaris.org/os/licensing.
12 # See the License for the specific language governing permissions
13 # and limitations under the License.
14 #
15 # When distributing Covered Code, include this CDDL HEADER in each
16 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
17 # If applicable, add the following below this CDDL HEADER, with the
18 # fields enclosed by brackets "[]" replaced with your own identifying
19 # information: Portions Copyright [yyyy] [name of copyright owner]
20 #
21 # CDDL HEADER END
22 #
23 #
24 # ident "%Z%M% %I% %E% SMI"
25 #
26 # Copyright 2005 Sun Microsystems, Inc. All rights reserved.
27 # Use is subject to license terms.
28 #
29 # SVM Flash cleanup
30 #
31 # Remove existing master SVM configuration on clone after full flash install.
32 # Restore existing clone SVM configuration after differential flash install.
33 #
34 #
35 #
36 IN_SYS=${FLASH_ROOT}/etc/system
37 IN_CONF=${FLASH_ROOT}/kernel/drv/md.conf
38 IN_CF=${FLASH_ROOT}/etc/lvm/mddb.cf
39 #
40 #
41 #
42 #
43 #
44 #
45 #
46 # Remove any SVM root entry from /etc/system file.
47 #
48 #
49 #
50 #
51 #
52 #
53 #
54 #
55 # Check if we are on the mini-root. If we are, we need to clean up the
56 # mddb configuration since this implies we are doing a full flash onto
57 # a fresh system.
58 #
```

new/usr/src/cmd/lvm/etc/svm.cleanup.sh

2

```
59 # If we are not on the mini-root that must mean we are installing
60 # the full flash via live-upgrade. In that case we share the
61 # SVM configuration with the currently running system so we
62 # need to copy the md.conf file from the current root onto the
63 # newly installed root. Note that the flash archive might not have
64 # been created from the currently running system.
65 if [ -h /kernel/drv/md.conf ]; then
66 # Remove SVM mddb entries from /kernel/drv/md.conf.
67 /usr/xpg4/bin/awk '
68 nawk '
69 BEGIN {delmddb=0}
70 /*# Begin MDD database info \((do not edit\)*/ {delmddb=1}
71 /*# End MDD database info \((do not edit\)*/ {delmddb=0; next}
72 {if (delmddb == 0) print $0}
73 ' ${IN_CONF} > ${TMP_CONF}
74 cp ${TMP_CONF} ${IN_CONF}
75 # Remove SVM mddb entries from /etc/lvm/mddb.cf.
76 /usr/xpg4/bin/awk '
77 nawk '
78 /*#/ {print $0}
79 ' ${IN_CF} > ${TMP_CF}
80 cp ${TMP_CF} ${IN_CF}
81 else
82 # copy SVM config from current root to new root
83 cp /kernel/drv/md.conf ${IN_CONF}
84 cp /etc/lvm/mddb.cf ${IN_CF}
85 fi
86 #
87 # We may need to enable the SVM services in SMF. This could happen
88 # if we used jumpstart or live-upgrade to create SVM volumes as
89 # part of the flash install.
90 #
91 # It doesn't matter if we are doing a flash install via a jumpstart
92 # on the mini-root or via a live-upgrade. In both cases we check
93 # the md.conf on the currently running root to see if SVM is
94 # configured. For the jumpstart case it will have setup the
95 # volumes already so the mini-root md.conf has the mddb info. For
96 # the live-upgrade case both roots will be sharing the same md.conf
97 # and have the same view of the SVM configuration.
98 #
99 # Check if there are mddb entries in md.conf to determine if SVM is
100 # configured.
101 sed -e 's/#.*$// ' /kernel/drv/md.conf | \
102 egrep '^[ ]*mddb_bootlist' >/dev/null 2>&1
103 MDDB_STATUS=?
104 #
105 if [ $MDDB_STATUS -eq 0 ]; then
106 echo "/usr/sbin/svcadm enable system/metainit:default" >> \
107 ${FLASH_ROOT}/var/svc/profile/upgrade
108 #
109 echo "/usr/sbin/svcadm enable system/mdmonitor:default" >> \
110 ${FLASH_ROOT}/var/svc/profile/upgrade
111 #
112 echo "/usr/sbin/svcadm enable network/rpc/meta:default" >> \
113 ${FLASH_ROOT}/var/svc/profile/upgrade
114 fi
115 #
116 else
117 # Differential flash install, restore clone SVM configuration.
118 # The matrix of master/clone SVM config w/ diff. flash looks like:
119 #
120 # master clone clone after differential flash
121 #
122 # yes yes same as clone prior to diff. flash
```



```

123 # yes no no
124 # no yes same as clone prior to diff. flash
125 # no no no
126 #

128 # restore saved config files
129 cp ${SAVE_DIR}/md.conf ${FLASH_ROOT}/kernel/drv/md.conf
130 cp ${SAVE_DIR}/devpath ${FLASH_ROOT}/etc/lvm/devpath
131 cp ${SAVE_DIR}/md.cf ${FLASH_ROOT}/etc/lvm/md.cf
132 cp ${SAVE_DIR}/md.ctrlmap ${FLASH_ROOT}/etc/lvm/md.ctrlmap
133 cp ${SAVE_DIR}/md.tab ${FLASH_ROOT}/etc/lvm/md.tab
134 cp ${SAVE_DIR}/mddb.cf ${FLASH_ROOT}/etc/lvm/mddb.cf
135 cp ${SAVE_DIR}/runtime.cf ${FLASH_ROOT}/etc/lvm/runtime.cf

137 # Now process the various permutations for the master and clone
138 # /etc/system file SVM root entries.

140 # First check if we need to do anything with /etc/system.
141 if `cmp -s ${SAVE_DIR}/system ${IN_SYS} >/dev/null 2>&1`; then
142     # There is no difference so leave it alone.
143     exit 0;
144 fi

146 # Get any SVM root entry from master /etc/system file.
147 MASTER_ROOT=`usr/xpg4/bin/awk '
148 MASTER_ROOT=`nawk '
149 BEGIN {inroot=0}
150 /\^* Begin MDD root info \(\do not edit\)/ {inroot=1; next}
151 /\^* End MDD root info \(\do not edit\)/ {inroot=0}
152 {if (inroot == 1) print $0}
153 ' ${IN_SYS}`

154 # Get any SVM root entry from clone /etc/system file.
155 CLONE_ROOT=`usr/xpg4/bin/awk '
156 CLONE_ROOT=`nawk '
157 BEGIN {inroot=0}
158 /\^* Begin MDD root info \(\do not edit\)/ {inroot=1; next}
159 /\^* End MDD root info \(\do not edit\)/ {inroot=0}
160 {if (inroot == 1) print $0}
161 ' ${SAVE_DIR}/system`

162 # If there is an SVM root entry in the master /etc/system file.
163 if [ "${MASTER_ROOT}" ]; then

165     # If there is an SVM root entry in the clone /etc/system file.
166     if [ "${CLONE_ROOT}" ]; then

168         # Restore clone SVM root entry in /etc/system file.
169         /usr/xpg4/bin/awk -v clone_root="${CLONE_ROOT}" '
170         nawk -v clone_root="${CLONE_ROOT}" '
171         BEGIN {newroot=0}
172         /\^* Begin MDD root info \(\do not edit\)/ {
173             newroot=1
174             print $0
175             print clone_root
176         }
177         /\^* End MDD root info \(\do not edit\)/ {newroot=0}
178         {if (newroot == 0) print $0}
179         ' ${IN_SYS} >${TMP_SYS}
180         cp ${TMP_SYS} ${IN_SYS}

181     else

183         # There is no SVM root entry in the clone so remove the entry
184         # from the /etc/system file.
185         /usr/xpg4/bin/awk '

```

```

187 nawk '
188 BEGIN {delroot=0}
189 /\^* Begin MDD root info \(\do not edit\)/ {delroot=1}
190 /\^* End MDD root info \(\do not edit\)/ {delroot=0; next }
191 {if (delroot == 0) print $0}
192 ' ${IN_SYS} >${TMP_SYS}
193 cp ${TMP_SYS} ${IN_SYS}

193 fi

195 else
196 # Master has no SVM root entry in the /etc/system file.
197 if [ "${CLONE_ROOT}" ]; then
198     # But clone does have one so we need to add it back in.

200     echo "* Begin MDD root info (do not edit)" >> ${IN_SYS}
201     echo "${CLONE_ROOT}" >> ${IN_SYS}
202     echo "* End MDD root info (do not edit)" >> ${IN_SYS}
203 fi

205 # If neither master nor clone has SVM root entry then
206 # we just leave the system file alone.
207 fi
208 fi

210 exit 0

```

new/usr/src/cmd/lvm/rpc.metamhd/i386/Makefile

1

```
*****
2600 Sun May 26 18:22:41 2013
new/usr/src/cmd/lvm/rpc.metamhd/i386/Makefile
XXX Remove nawk(1)
*****
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 1996, 2001-2003 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # ident "%Z%M% %I% %E% SMI"
27 #
26 # Makefile for logical volume management
27 #

29 PROG= rpc.metamhd

31 RPCMOD = metamhd

33 DERIVED_FILES = \
34     $(RPCMOD)_svc.c \
35     $(RPCMOD)_xdr.c \
36     mhd_xdr.c

38 OBJECTS= \
39     mhd_drive.o \
40     mhd_error.o \
41     mhd_failfast.o \
42     mhd_freeresult.o \
43     mhd_init.o \
44     mhd_mem.o \
45     mhd_metamhd.o \
46     mhd_set.o \
47     mhd_synch.o \
48     mhd_time.o

50 LINTOBJECTS= \
51     mhd_drive.o \
52     mhd_error.o \
53     mhd_failfast.o \
54     mhd_freeresult.o \
55     mhd_init.o \
56     mhd_mem.o \
57     mhd_metamhd.o \
58     mhd_set.o \
59     mhd_synch.o \
```

new/usr/src/cmd/lvm/rpc.metamhd/i386/Makefile

2

```
60         mhd_time.o

62 OBJECTS += $(DERIVED_FILES:.c=.o)

64 SRCS = $(OBJECTS:%.o=./%.c)
65 LINTSRCS = $(LINTOBJECTS:%.o=./%.c)

67 include ../../Makefile.cmd
68 include ../../Makefile.lvm

71 MDLIBLES =
72 LDLIBS +=      -ladm -lsocket -lnsl

74 POFILES= $(OBJECTS:%.o=%.po)

77 $(RPCMOD)_svc.c := RPCGENFLAGS += -A -K -l

79 DEFINES += -D_REENTRANT
80 CFLAGS += $(DEFINES)

83 #
84 #
85 lint := LINTFLAGS += -m

87 .KEEP_STATE:

89 %.o:    ../%.c
90         $(COMPILE.c) $<

92 all:    $(PROG)

94 $(PROG): $(OBJECTS)
95         $(LINK.c) -o $@ $(OBJECTS) $(LDLIBS)
96         $(POST_PROCESS)

99 ROOTUSRSBINPROG=$(PROG:%=$(ROOTUSRSBIN)/%)
100 install: all $(ROOTUSRSBINPROG)

102 catalog:

104 cstyle:
105         ${CSTYLE} ${SRCS}

107 lint:
108         ${LINT.c} $(LINTFLAGS) ${LINTSRCS}

110 clean:
111         ${RM} ${OBJECTS} ${DERIVED_FILES} *.o a.out core metamhd.x

113 clobber: clean
114         ${RM} $(PROG)

116 metamhd_svc.c: $(SRC)/head/metamhd.x
117         $(CP) $(SRC)/head/metamhd.x .
118         $(RPCGEN) $(RPCGENFLAGS_SERVER) metamhd.x -o $@
119         ${RM} metamhd.x

121 metamhd_xdr.c: $(SRC)/head/metamhd.x
122         $(RPCGEN) $(RPCGENFLAGS) -c $(SRC)/head/metamhd.x -o $@

124 mhd_xdr.c: $(SRC)/uts/common/sys/lvm/mhd_x
125         $(RPCGEN) $(RPCGENFLAGS) -c $(SRC)/uts/common/sys/lvm/mhd_x | \
```

new/usr/src/cmd/lvm/rpc.metamhd/i386/Makefile

3

```
126 /usr/xpg4/bin/awk '{sub(/uts/common/sys/lvm/, "head"); print $$0}' >  
128 nawk '{sub(/uts/common/sys/lvm/, "head"); print $$0}' >${@}
```

new/usr/src/cmd/lvm/rpc.metamhd/sparc/Makefile

1

```
*****
2600 Sun May 26 18:22:42 2013
new/usr/src/cmd/lvm/rpc.metamhd/sparc/Makefile
XXX Remove nawk(1)
*****
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 1996, 2001-2003 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # ident "%Z%M% %I% %E% SMI"
27 #
26 # Makefile for logical volume management
27 #

29 PROG= rpc.metamhd

31 RPCMOD = metamhd

33 DERIVED_FILES = \
34     $(RPCMOD)_svc.c \
35     $(RPCMOD)_xdr.c \
36     mhd_xdr.c

38 OBJECTS= \
39     mhd_drive.o \
40     mhd_error.o \
41     mhd_failfast.o \
42     mhd_freeresult.o \
43     mhd_init.o \
44     mhd_mem.o \
45     mhd_metamhd.o \
46     mhd_set.o \
47     mhd_synch.o \
48     mhd_time.o

50 LINTOBJECTS= \
51     mhd_drive.o \
52     mhd_error.o \
53     mhd_failfast.o \
54     mhd_freeresult.o \
55     mhd_init.o \
56     mhd_mem.o \
57     mhd_metamhd.o \
58     mhd_set.o \
59     mhd_synch.o \
```

new/usr/src/cmd/lvm/rpc.metamhd/sparc/Makefile

2

```
60     mhd_time.o

62 OBJECTS += $(DERIVED_FILES:.c=.o)

64 SRCS = $(OBJECTS:%.o=./%.c)
65 LINTSRCS = $(LINTOBJECTS:%.o=./%.c)

67 include ../../Makefile.cmd
68 include ../../Makefile.lvm

71 MDLIBLES =
72 LDLIBS += -ladm -lsocket -lnsl

74 POFILES= $(OBJECTS:%.o=%.po)

77 $(RPCMOD)_svc.c := RPCGENFLAGS += -A -K -l

79 DEFINES += -D_REENTRANT
80 CFLAGS += $(DEFINES)

83 #
84 #
85 lint := LINTFLAGS += -m

87 .KEEP_STATE:

89 %.o: ../%.c
90     $(COMPILE.c) $<

92 all: $(PROG)

94 $(PROG): $(OBJECTS)
95     $(LINK.c) -o $@ $(OBJECTS) $(LDLIBS)
96     $(POST_PROCESS)

99 ROOTUSRSBINPROG=$(PROG:%=$(ROOTUSRSBIN)/%)
100 install: all $(ROOTUSRSBINPROG)

102 catalog:

104 cstyle:
105     ${CSTYLE} ${SRCS}

107 lint:
108     ${LINT.c} ${LINTFLAGS} ${LINTSRCS}

110 clean:
111     ${RM} ${OBJECTS} ${DERIVED_FILES} *.o a.out core metamhd.x

113 clobber: clean
114     ${RM} $(PROG)

116 metamhd_svc.c: $(SRC)/head/metamhd.x
117     $(CP) $(SRC)/head/metamhd.x .
118     $(RPCGEN) $(RPCGENFLAGS_SERVER) metamhd.x -o $@
119     ${RM} metamhd.x

121 metamhd_xdr.c: $(SRC)/head/metamhd.x
122     $(RPCGEN) $(RPCGENFLAGS) -c $(SRC)/head/metamhd.x -o $@

124 mhd_xdr.c: $(SRC)/uts/common/sys/lvm/mhd_x
125     $(RPCGEN) $(RPCGENFLAGS) -c $(SRC)/uts/common/sys/lvm/mhd_x | \
```

new/usr/src/cmd/lvm/rpc.metamhd/sparc/Makefile

3

```
126 /usr/xpg4/bin/awk '{sub(/uts/common/sys/lvm/, "head"); print $$0}' >  
128 nawk '{sub(/uts/common/sys/lvm/, "head"); print $$0}' >${@}
```

new/usr/src/cmd/lvm/util/svc-metasync

1

```
*****
3582 Sun May 26 18:22:42 2013
new/usr/src/cmd/lvm/util/svc-metasync
XXX Remove awk(1)
*****
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright 2006 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # ident "%Z%M% %I% %E% SMI"
27 #
28 # Start mirror resync threads.

28 DEVFSADM=/usr/sbin/devfsadm
29 METADEVADM=/usr/sbin/metadevadm
30 METASYNC=/usr/sbin/metasync
31 METADEV=/dev/md/admin
32 METASET=/usr/sbin/metaset
33 TMPFILE=/var/run/metaset.$$

35 . /lib/svc/share/smf_include.sh

37 print_verbose()
38 {
39     echo "Unable to resolve unnamed devices for volume management."
40     echo "Please refer to the Solaris Volume Manager documentation,"
41     echo "Troubleshooting section, at http://docs.sun.com or from"
42     echo "your local copy."
43 }

45 resolve_auto_take_sets()
46 {
47     if [ -x $METASET ]; then
48         # Fixing up of the ctd names for devices in auto take
49         # sets relies heavily on the output of the metaset
50         # command. Any change to the output of the metaset command
51         # should modify this script as well in order ensure nothing
52         # breaks
53         #
54         # The following command saves all of the auto-take set names
55         # into the TMPFILE
56         name_str=`gettext "Set name"`
57         mn_str=`gettext "Multi-owner"`
58         $METASET | /usr/xpg4/bin/awk -F ' |\t|,' -v snm="$name_str" \
60         $METASET | /bin/nawk -F ' |\t|,' -v snm="$name_str" \
```

new/usr/src/cmd/lvm/util/svc-metasync

2

```
59         -v mstr="$mn_str" '$0 ~ snm { \
60         if (index($0, mstr) == 0) print $4 \
61         }' > $TMPFILE 2>&1

63     if [ -s "$TMPFILE" ]; then
64         localised_string=`gettext "Yes (auto)"`
65         for i in `cat $TMPFILE`; do
66             $METASET -s $i | grep "$localised_string" \
67             > /dev/null 2>&1
68             if [ $? -eq 0 ]; then
69                 $METADEVADM -l -r -s $i
70                 error=$?
71                 case $error in
72                 0|2) ;;
73                 3) print_verbose
74                     ;;
75                 *) echo "$METADEVADM \
76                     -r failure $error."
77                     ;;
78                 esac
79             fi
80         done
81     fi
82     if [ -f "$TMPFILE" ]; then
83         /usr/bin/rm -f $TMPFILE
84     fi
85 fi
86 }
unchanged_portion_omitted_
```

```

*****
6742 Sun May 26 18:22:43 2013
new/usr/src/cmd/mdb/tools/scripts/hdr2map.sh
XXX Remove nawk(1)
*****
1 #!/bin/ksh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright 2006 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #ident "%Z%M% %I% %E% SMI"
27 #

27 #
28 # Given a header file, extract function prototypes and global variable
29 # declarations in a form that can be used in a mapfile. The list of extracted
30 # functions and variables will be combined with a user-specified template to
31 # create a complete mapfile.
32 #
33 # Template
34 # -----
35 #
36 # The template contains two sections - the prologue, and the epilogue. These
37 # sections are used, verbatim, as the beginning and the end of the mapfile.
38 # Sections begin and end with single-line comments whose sole contents are
39 # "/* BEGIN $section */" and "/* END $section */".
40 #
41 # Template example:
42 #
43 # /* BEGIN PROLOGUE */
44 # [ ... prologue goes here ... ]
45 # /* END PROLOGUE */
46 # /* BEGIN EPILOGUE */
47 # [ ... epilogue goes here ... ]
48 # /* END EPILOGUE */
49 #
50 # Selective Exportation
51 # -----
52 #
53 # Some header files will have a public/private interface mix that is strongly
54 # biased towards private interfaces. That is, of the interfaces declared by
55 # a given header file, the majority of them are private. Only a small subset
56 # of interfaces are to be exported publicly. Using Selective Exportation, a
57 # special comment is included in the header file, declaring to this script that
58 # only a subset of interfaces - those with a marking declared in the comment -
59 # should be included in the mapfile. The marking is itself a special comment,

```

```

60 # whose format is declared using a directive like this:
61 #
62 #     MAPFILE: export "Driver OK"
63 #
64 # Using the above directive, only those function prototypes and variable
65 # declarations with "/* Driver OK */" comments included in the mapfile. Note
66 # that the comment must be at the end of the first line. If the declaration
67 # spans multiple lines, the exportation comment must appear on the first line.
68 #
69 # Examples of functions selected for exportation:
70 #
71 # MAPFILE: export "Driver OK"
72 #
73 # extern int foo(int);           /* Driver OK */
74 # extern void bar(int, int,     /* Driver OK */
75 #             int, void *);
76 #
77 # Selective Exportation may not be used in the same file as Selective Exclusion.
78 #
79 # Selective Exclusion
80 # -----
81 #
82 # Selective Exclusion is to be used in cases where the public/private interface
83 # mix is reversed - where public interfaces greatly outnumber the private ones.
84 # In this case, we want to be able to mark the private ones, thus telling this
85 # script that the marked interfaces are to be excluded from the mapfile.
86 # Marking is accomplished via a process similar to that used for Selective
87 # Exportation. A directive is included in a comment, and is formatted like
88 # this:
89 #
90 #     MAPFILE: exclude "Internal"
91 #
92 # Using the above directive, function prototypes and variable declarations with
93 # "/* Internal */" comments would be excluded. Note that the comment must be at
94 # the end of the first line. If the declaration spans multiple lines, the
95 # exclusion comment must appear on the first line.
96 #
97 # Examples of functions excluded from exportation:
98 #
99 # MAPFILE: exclude "Internal"
100 #
101 # extern int foo(int);           /* Internal */
102 # extern void bar(int, int,     /* Internal */
103 #             int, void *);
104 #
105 # Selective Exclusion may not be used in the same file as Selective Exportation.
106 #

108 function extract_prototypes
109 {
110     typeset header="$1"
111     typeset prefix="$2"
112
113     /usr/xpg4/bin/awk -v prefix="$prefix" <$header '
114     nawk -v prefix="$prefix" <$header '
115     /^.*MAPFILE: export \"[^\"]*\"$/ {
116         if (protoexclude) {
117             print "ERROR: export after exclude\n";
118             exit(1);
119         }
120     }
121     sub(/^[^\"]*\"/, "");
122     sub(/\"$/, "");

123     exportmark=sprintf("/* %s */", $0);
124     next;

```

```

125     }
127     /^.*MAPFILE: exclude \"[^\"]*\"$/ {
128         if (protomatch) {
129             print "ERROR: exclude after export";
130             exit(1);
131         }
133         sub(/^\"[^\"]*\"/, "");
134         sub(/\"$/ , "");
136         excludemark=sprintf("/ * %s */", $0);
137         next;
138     }
140     exportmark {
141         # Selective Exportation has been selected (exportmark is
142         # set), so exclude this line if it does not have the
143         # magic export mark.
144         if (length($0) < length(exportmark) ||
145             substr($0, length($0) - length(exportmark) + 1) != \
146             exportmark)
147             next;
148     }
150     excludemark {
151         # Selective Exclusion has been selected (excludemark is
152         # set), so exclude this line only if it has the magic
153         # exclude mark.
154         if (length($0) > length(excludemark) &&
155             substr($0, \
156                 length($0) - length(excludemark) + 1) == \
157             excludemark)
158             next;
159     }
161     # Functions
162     /^extern.*\(/ {
163         for (i = 1; i <= NF; i++) {
164             if (sub(/\".*$/, "", $i)) {
165                 sub(/^\"/, "", $i);
166                 if (!seenfn[$i]) {
167                     printf("%s%s;\n", prefix, $i);
168                     seenfn[$i] = 1;
169                 }
170                 break;
171             }
172         }
173         next;
174     }
176     # Global variables
177     /^extern[^\(\)]*/ {
178         for (i = 1; i <= NF; i++) {
179             if (match($i, /;/)) {
180                 printf("%s%s; /* variable */\n", prefix,
181                     substr($i, 1, length($i) - 1));
182                 break;
183             }
184         }
185         next;
186     }
187     ' || die "Extraction failed"
188 }

```

```
190 function extract_section
```

```

191 {
192     typeset skel="$1"
193     typeset secname="$2"
195     /usr/xpg4/bin/awk <$skel -v name=$secname -v skel=$skel '
197     nawk <$skel -v name=$secname -v skel=$skel '
196     /\\"* [^ ]* [^ ]* \\"// && $3 == name {
197         if ($2 == "BEGIN") {
198             printing = 1;
199         } else {
200             printing = 0;
201         }
202         next;
203     }
205     printing != 0 { print; }
206 }
207 }

```

unchanged_portion_omitted


```
new/usr/src/cmd/mdb/tools/scripts/map2linktest.sh
```

1

```
*****
2539 Sun May 26 18:22:43 2013
new/usr/src/cmd/mdb/tools/scripts/map2linktest.sh
XXX Remove awk(1)
*****
1 #!/bin/ksh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright 2006 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #ident "%Z%M% %I% %E% SMI"
27 #
28 #
29 # Create dummy functions for each of the functions in the module API. We can
30 # then link a module against an object file created from the output of this
31 # script to determine whether or not that module restricts itself to the API.
32 # If the module uses functions outside of the module API, then it cannot be
33 # used as a kmdb module.
34 #
35 # /usr/xpg4/bin/awk '
36 # awk '
37 #
38 #
39 #
40 #
41 #
42 #
43 #
44 #
45 # Skip blank lines and comments
46 # /^$/ { next; }
47 # /^[ \t]*##/ { next; }
48 #
49 # Print globals only
50 # printing == 0 { next; }
51 #
52 # Symbols beginning with "kmdb_" are not in the module API - they are
53 # private to kmdb.
54 # $1 ~ /^kmdb_/ { next; }
55 #
56 # Symbols which have the token "variable" are seen as an int
57 # $3 ~ /variable/ {
58 #     if (seen[$1]) {
```

```
new/usr/src/cmd/mdb/tools/scripts/map2linktest.sh
```

2

```
59         next;
60     }
61 #
62     seen[$1] = 1;
63 #
64     printf("int %s = 0;\n", substr($1, 1, length($1) - 1));
65     next;
66 }
67 #
68 $1 !~ /^[ \t]*$/ { next; }
69 #
70 # Print everything else that we have not already seen as a function
71 # definition so we can create our filter.
72 {
73     if (seen[$1]) {
74         next;
75     }
76 #
77     seen[$1] = 1;
78 #
79     printf("void %s(void) {}\n", substr($1, 1, length($1) - 1));
80 }
81 '
82 #
83 #
84 # kmdb modules cannot have their own _init, _fini, or _info routines. By
85 # creating dummies for them here, a link against an object file created from
86 # the output of this script will fail if the module defines one of them.
87 #
88 echo "void _init(void) {}"
89 echo "void _info(void) {}"
90 echo "void _fini(void) {}"
91 #
92 # The SunStudio compiler may generate calls to _memcpy and so we
93 # need to make sure that the correct symbol exists for these calls.
94 #
95 echo "void _memcpy(void) {}"
```

```
new/usr/src/cmd/mdb/tools/scripts/tigen.sh
```

1

```
*****
```

```
4212 Sun May 26 18:22:43 2013
```

```
new/usr/src/cmd/mdb/tools/scripts/tigen.sh
```

```
XXX Remove awk(1)
```

```
*****
```

```
_____unchanged_portion_omitted_
```

```
52 extract_section()
```

```
53 {
```

```
54     typeset skel="$1"
```

```
55     typeset secname="$2"
```

```
57     /usr/xpg4/bin/awk <$$skel -v name=$secname -v skel=$skel '
```

```
57     awk <$$skel -v name=$secname -v skel=$skel '
```

```
58     /\[/\* [^\ ]* [^\ ]* \[/\// && $3 == name {
```

```
59         if ($2 == "BEGIN") {
```

```
60             printing = 1;
```

```
61             printf("# %d \"%s\"\n", NR + 1, skel);
```

```
62         } else {
```

```
63             printing = 0;
```

```
64         }
```

```
65         next;
```

```
66     }
```

```
68     , printing != 0 { print; }
```

```
69
```

```
70 }
```

```
_____unchanged_portion_omitted_
```

new/usr/src/cmd/print/scripts/conv_lpd

1

```
*****
4827 Sun May 26 18:22:44 2013
new/usr/src/cmd/print/scripts/conv_lpd
XXX Remove awk(1)
*****
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License, Version 1.0 only
7 # (the "License"). You may not use this file except in compliance
8 # with the License.
9 #
10 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
11 # or http://www.opensolaris.org/os/licensing.
12 # See the License for the specific language governing permissions
13 # and limitations under the License.
14 #
15 # When distributing Covered Code, include this CDDL HEADER in each
16 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
17 # If applicable, add the following below this CDDL HEADER, with the
18 # fields enclosed by brackets "[]" replaced with your own identifying
19 # information: Portions Copyright [yyyy] [name of copyright owner]
20 #
21 # CDDL HEADER END
22 #
23 #
24 # Copyright (c) 1994, 1995, 1996 by Sun Microsystems, Inc.
25 # All Rights Reserved
26 #
27 # ident "%Z%M% %I% %E% SMI"
28 #
29 # Printcap <-> Printers.conf conversion utility...
30 #
31 #
32 # Usage: conv_lpd [ -c (printers|printcap) ] [ -n ] (file)
33 #
34 #
35 #
36 #
37 #
38 #
39 #
40 #
41 #
42 #
43 #
44 #
45 #
46 #
47 #
48 #
49 #
50 #
51 #
52 #
53 #
54 #
55 #
56 #
57 #
58 #
59 #
60 #
61 #
62 #
63 #
64 #
65 #
66 #
67 #
68 #
69 #
70 #
71 #
72 #
73 #
74 #
75 #
76 #
77 #
78 #
79 #
80 #
81 #
82 #
83 #
84 #
85 #
86 #
87 #
88 #
89 #
90 #
91 #
92 #
93 #
94 #
95 #
96 #
97 #
98 #
99 #
100 #
101 #
102 #
103 #
104 #
105 #
106 #
107 #
108 #
109 #
110 #
111 #
112 #
113 #
114 #
115 #
116 #
117 #
118 #
119 #
120 #
121 #
122 #
123 #
124 #
125 #
126 #
127 #
128 #
129 #
130 #
131 #
132 #
133 #
134 #
135 #
136 #
137 #
138 #
139 #
140 #
141 #
142 #
143 #
144 #
145 #
146 #
147 #
148 #
149 #
150 #
151 #
152 #
153 #
154 #
155 #
156 #
157 #
158 #
159 #
160 #
161 #
162 #
163 #
164 #
165 #
166 #
167 #
168 #
169 #
170 #
171 #
172 #
173 #
174 #
175 #
176 #
177 #
178 #
179 #
180 #
181 #
182 #
183 #
184 #
185 #
186 #
187 #
188 #
189 #
190 #
191 #
192 #
193 #
194 #
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #
221 #
222 #
223 #
224 #
225 #
226 #
227 #
228 #
229 #
230 #
231 #
232 #
233 #
234 #
235 #
236 #
237 #
238 #
239 #
240 #
241 #
242 #
243 #
244 #
245 #
246 #
247 #
248 #
249 #
250 #
251 #
252 #
253 #
254 #
255 #
256 #
257 #
258 #
259 #
260 #
261 #
262 #
263 #
264 #
265 #
266 #
267 #
268 #
269 #
270 #
271 #
272 #
273 #
274 #
275 #
276 #
277 #
278 #
279 #
280 #
281 #
282 #
283 #
284 #
285 #
286 #
287 #
288 #
289 #
290 #
291 #
292 #
293 #
294 #
295 #
296 #
297 #
298 #
299 #
300 #
301 #
302 #
303 #
304 #
305 #
306 #
307 #
308 #
309 #
310 #
311 #
312 #
313 #
314 #
315 #
316 #
317 #
318 #
319 #
320 #
321 #
322 #
323 #
324 #
325 #
326 #
327 #
328 #
329 #
330 #
331 #
332 #
333 #
334 #
335 #
336 #
337 #
338 #
339 #
340 #
341 #
342 #
343 #
344 #
345 #
346 #
347 #
348 #
349 #
350 #
351 #
352 #
353 #
354 #
355 #
356 #
357 #
358 #
359 #
360 #
361 #
362 #
363 #
364 #
365 #
366 #
367 #
368 #
369 #
370 #
371 #
372 #
373 #
374 #
375 #
376 #
377 #
378 #
379 #
380 #
381 #
382 #
383 #
384 #
385 #
386 #
387 #
388 #
389 #
390 #
391 #
392 #
393 #
394 #
395 #
396 #
397 #
398 #
399 #
400 #
401 #
402 #
403 #
404 #
405 #
406 #
407 #
408 #
409 #
410 #
411 #
412 #
413 #
414 #
415 #
416 #
417 #
418 #
419 #
420 #
421 #
422 #
423 #
424 #
425 #
426 #
427 #
428 #
429 #
430 #
431 #
432 #
433 #
434 #
435 #
436 #
437 #
438 #
439 #
440 #
441 #
442 #
443 #
444 #
445 #
446 #
447 #
448 #
449 #
450 #
451 #
452 #
453 #
454 #
455 #
456 #
457 #
458 #
459 #
460 #
461 #
462 #
463 #
464 #
465 #
466 #
467 #
468 #
469 #
470 #
471 #
472 #
473 #
474 #
475 #
476 #
477 #
478 #
479 #
480 #
481 #
482 #
483 #
484 #
485 #
486 #
487 #
488 #
489 #
490 #
491 #
492 #
493 #
494 #
495 #
496 #
497 #
498 #
499 #
500 #
501 #
502 #
503 #
504 #
505 #
506 #
507 #
508 #
509 #
510 #
511 #
512 #
513 #
514 #
515 #
516 #
517 #
518 #
519 #
520 #
521 #
522 #
523 #
524 #
525 #
526 #
527 #
528 #
529 #
530 #
531 #
532 #
533 #
534 #
535 #
536 #
537 #
538 #
539 #
540 #
541 #
542 #
543 #
544 #
545 #
546 #
547 #
548 #
549 #
550 #
551 #
552 #
553 #
554 #
555 #
556 #
557 #
558 #
559 #
560 #
561 #
562 #
563 #
564 #
565 #
566 #
567 #
568 #
569 #
570 #
571 #
572 #
573 #
574 #
575 #
576 #
577 #
578 #
579 #
580 #
581 #
582 #
583 #
584 #
585 #
586 #
587 #
588 #
589 #
590 #
591 #
592 #
593 #
594 #
595 #
596 #
597 #
598 #
599 #
600 #
601 #
602 #
603 #
604 #
605 #
606 #
607 #
608 #
609 #
610 #
611 #
612 #
613 #
614 #
615 #
616 #
617 #
618 #
619 #
620 #
621 #
622 #
623 #
624 #
625 #
626 #
627 #
628 #
629 #
630 #
631 #
632 #
633 #
634 #
635 #
636 #
637 #
638 #
639 #
640 #
641 #
642 #
643 #
644 #
645 #
646 #
647 #
648 #
649 #
650 #
651 #
652 #
653 #
654 #
655 #
656 #
657 #
658 #
659 #
660 #
661 #
662 #
663 #
664 #
665 #
666 #
667 #
668 #
669 #
670 #
671 #
672 #
673 #
674 #
675 #
676 #
677 #
678 #
679 #
680 #
681 #
682 #
683 #
684 #
685 #
686 #
687 #
688 #
689 #
690 #
691 #
692 #
693 #
694 #
695 #
696 #
697 #
698 #
699 #
700 #
701 #
702 #
703 #
704 #
705 #
706 #
707 #
708 #
709 #
710 #
711 #
712 #
713 #
714 #
715 #
716 #
717 #
718 #
719 #
720 #
721 #
722 #
723 #
724 #
725 #
726 #
727 #
728 #
729 #
730 #
731 #
732 #
733 #
734 #
735 #
736 #
737 #
738 #
739 #
740 #
741 #
742 #
743 #
744 #
745 #
746 #
747 #
748 #
749 #
750 #
751 #
752 #
753 #
754 #
755 #
756 #
757 #
758 #
759 #
760 #
761 #
762 #
763 #
764 #
765 #
766 #
767 #
768 #
769 #
770 #
771 #
772 #
773 #
774 #
775 #
776 #
777 #
778 #
779 #
780 #
781 #
782 #
783 #
784 #
785 #
786 #
787 #
788 #
789 #
790 #
791 #
792 #
793 #
794 #
795 #
796 #
797 #
798 #
799 #
800 #
801 #
802 #
803 #
804 #
805 #
806 #
807 #
808 #
809 #
810 #
811 #
812 #
813 #
814 #
815 #
816 #
817 #
818 #
819 #
820 #
821 #
822 #
823 #
824 #
825 #
826 #
827 #
828 #
829 #
830 #
831 #
832 #
833 #
834 #
835 #
836 #
837 #
838 #
839 #
840 #
841 #
842 #
843 #
844 #
845 #
846 #
847 #
848 #
849 #
850 #
851 #
852 #
853 #
854 #
855 #
856 #
857 #
858 #
859 #
860 #
861 #
862 #
863 #
864 #
865 #
866 #
867 #
868 #
869 #
870 #
871 #
872 #
873 #
874 #
875 #
876 #
877 #
878 #
879 #
880 #
881 #
882 #
883 #
884 #
885 #
886 #
887 #
888 #
889 #
890 #
891 #
892 #
893 #
894 #
895 #
896 #
897 #
898 #
899 #
900 #
901 #
902 #
903 #
904 #
905 #
906 #
907 #
908 #
909 #
910 #
911 #
912 #
913 #
914 #
915 #
916 #
917 #
918 #
919 #
920 #
921 #
922 #
923 #
924 #
925 #
926 #
927 #
928 #
929 #
930 #
931 #
932 #
933 #
934 #
935 #
936 #
937 #
938 #
939 #
940 #
941 #
942 #
943 #
944 #
945 #
946 #
947 #
948 #
949 #
950 #
951 #
952 #
953 #
954 #
955 #
956 #
957 #
958 #
959 #
960 #
961 #
962 #
963 #
964 #
965 #
966 #
967 #
968 #
969 #
970 #
971 #
972 #
973 #
974 #
975 #
976 #
977 #
978 #
979 #
980 #
981 #
982 #
983 #
984 #
985 #
986 #
987 #
988 #
989 #
990 #
991 #
992 #
993 #
994 #
995 #
996 #
997 #
998 #
999 #
1000 #
```

new/usr/src/cmd/print/scripts/conv_lpd

2

```
60 TMPF1=/tmp/tinput1.$$
61 TMPF2=/tmp/tinput2.$$
62 FILE=/tmp/input.$$

64 # Any remaining arg is the "file" specification. It is a required arg.
65 if [ -z "$1" ]; then
66     echo $USAGE
67     exit 1
68 else
69     cp $1 $TMPF1
70 fi

72 echo >>$TMPF1
73 echo "_done" >>$TMPF1

75 #
76 # First, strip all continuation characters, leaving one, single line
77 # for each printer entry.
78 #
79     CONV_FIX=/usr/lib/print/conv_fix

81     if [ -f $CONV_FIX ]; then
82         $CONV_FIX -f $TMPF1 -o $TMPF2
83         if [ $? != 0 ]; then
84             echo "$0:"
85             gettext "Fatal Error: $CONV_FIX failed.\n"
86             gettext "Please contact your Sun support representative."
87             exit 1
88         fi
89     else
90         gettext "$0: Fatal: Cannot locate $CONV_FIX binary.\n"
91         gettext "Please contact your Sun support representative.\n"
92         exit 1
93     fi

95 #
96 # Continuation characters are now stripped. Continue processing.
97 #
98     /bin/sed -e "s/:[ \t]*:/:/g" $TMPF2 > $FILE

100 #
101 # Empty colons ":[ <TAB>]*:" are now stripped. Continue processing.
102 #

104 /usr/xpg4/bin/awk '
106     awk '
107     BEGIN {
108         "uname -n" | getline ;
109         host = $0 ;
110         found = 0 ;
111         local_pr = 0 ;
112     }
113     unchanged_portion_omitted_
```

new/usr/src/cmd/print/scripts/getmakes

1

1149 Sun May 26 18:22:44 2013

new/usr/src/cmd/print/scripts/getmakes

XXX Remove awk(1)

```
1 #!/usr/bin/ksh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
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 #
27 #
28 #
29 #
30 [[ -f /var/lp/ppd/ppdcache ]] || exit 1
31 /bin/cat /var/lp/ppd/ppdcache |
32 /usr/xpg4/bin/awk '{FS=":"; print $1}' |
33 /usr/xpg4/bin/awk '{print $1}' |
34 awk '{FS=":"; print $1}' |
35 awk '{print $1}' |
36 /bin/sort -u
37 exit 0
```

new/usr/src/cmd/print/scripts/getmodels

1

1328 Sun May 26 18:22:45 2013

new/usr/src/cmd/print/scripts/getmodels

XXX Remove awk(1)

```
1 #!/usr/bin/ksh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
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 #
27 # get a list of the Models for this Model from the ppcache
28 #
29 # Input:
30 #     Manufacturer      (Note: printmgr passes only the first word
31 #     Lexmark           of the manufacturer name)
32 # Output:
33 #     Manufacturer Model
34 #     Lexmark Optra Color 1200

36 if [[ $# -lt 1 ]]; then
37     exit 1
38 fi

40 cachefile=/var/lp/ppd/ppdcache
41 [[ -f $cachefile ]] || exit 1
42 /bin/grep "^$1" $cachefile | /usr/xpg4/bin/awk '{FS=":"; print $2}' | uniq
44 /bin/grep "^$1" $cachefile | awk '{FS=":"; print $2}' | uniq
43 exit 0
```

new/usr/src/cmd/print/scripts/getppdfile

1

```
*****
2451 Sun May 26 18:22:45 2013
new/usr/src/cmd/print/scripts/getppdfile
XXX Remove awk(1)
*****
1 #!/usr/bin/ksh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
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 #
27 # Get the path/ppdfilename for this ppd NickName
28 # Input:
29 #     make: model: ppdlabel: ppd:
30 #     PrintersRus: ABC Model 1234: SUNWfoomatic(S): Foomatic/Postscript (recom
31 #

33 #
34 # Returns the full path to the repository associated with
35 # the repository letter found between parenthesis in the
36 # extended PPD label.
37 #
38 # $1 - Extended PPD label
39 #
40 rep_path()
41 {
42     case "$(expr \"$1\" : \"*(\\(.*)\").*)\" in
43     \"S\")
44         echo \"/usr/share/ppd\"
45         ;;
46     \"V\")
47         echo \"/opt/share/ppd\"
48         ;;
49     \"A\")
50         echo \"/usr/local/share/ppd\"
51         ;;
52     \"U\")
53         echo \"/var/lp/ppd\"
54         ;;
55     esac
56 }

58 if [[ $# -lt 4 ]]; then
59     exit 1
```

new/usr/src/cmd/print/scripts/getppdfile

2

```
60 fi

62 [[ -f /var/lp/ppd/ppdcache ]] || exit 1
63 make=$(echo $* | /usr/xpg4/bin/awk '{FS=":"; print $1}')
65 make=$(echo $* | /usr/bin/nawk '{FS=":"; print $1}')
64 # strip leading blanks
65 model=$(echo $* | /usr/xpg4/bin/awk '{FS=":"; print $2}' |
67 model=$(echo $* | /usr/bin/nawk '{FS=":"; print $2}' |
66 /bin/sed -e 's/^[ ]*//')
67 extppdlabel=$(echo $* | /usr/xpg4/bin/awk '{FS=":"; print $3}' |
69 extppdlabel=$(echo $* | /usr/bin/nawk '{FS=":"; print $3}' |
68 /bin/sed -e 's/^[ ]*//')
69 ppd=$(echo $* | /usr/xpg4/bin/awk '{FS=":"; print $4}' |
71 ppd=$(echo $* | /usr/bin/nawk '{FS=":"; print $4}' |
70 /bin/sed -e 's/^[ ]*//')

72 #
73 # Do not use ":" with $make. printmgr collapses manufacturer name
74 # to first word, ie PrintersRus and PrintersRus International become
75 # PrintersRus. The full path to the PPD file will be the 6th
76 # colon separated entry in the ppdcache entry. If the format
77 # of a ppdcache entry changes, then this will need to be modified
78 # also.
79 #
80 /bin/grep "${make}" /var/lp/ppd/ppdcache |
81 /bin/grep "${model}:" |
82 /bin/grep "${ppd}:" |
83 /bin/grep "${rep_path ${extppdlabel}}/${extppdlabel%\"(*)\" |
84 /usr/xpg4/bin/awk '{FS=":"; print $6}'
86 /usr/bin/nawk '{FS=":"; print $6}'

86 exit 0
```

new/usr/src/cmd/print/scripts/ppdfilename2mmp

1

```
*****
2450 Sun May 26 18:22:46 2013
new/usr/src/cmd/print/scripts/ppdfilename2mmp
XXX Remove awk(1)
*****
1 #!/usr/bin/ksh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
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 #
27 # Get the make/model/nickname as well as the repository/label from ppdfilename
28 #

30 # Input
31 #   ppdfilename
32 #   /var/lp/ppd/user/HP/foo.ppd.gz
33 # Output
34 #   make
35 #   model
36 #   label(repository letter): driver
37 #
38 #   Lexmark
39 #   IBM Page Printer 3112
40 #   foomatic(L): Foomatic/hpijs
41 #

43 if [[ $# -lt 1 ]]; then
44     exit 1
45 fi

47 cachefile=/var/lp/ppd/ppdcache
48 [[ -f $cachefile ]] || exit 1

50 cacheentry=$(/bin/grep "$1" $cachefile)
51 [[ -n "$cacheentry" ]] || exit 1

53 #
54 # Retrieve the manufacturer (make)
55 # Use only the first word in manufacturer entry
56 #
57 manuf=$(echo "$cacheentry" |
58 /usr/xpg4/bin/awk '{FS=" "; print $1}' |
59 /usr/xpg4/bin/awk '{print $1}')
```

new/usr/src/cmd/print/scripts/ppdfilename2mmp

2

```
60 awk '{FS=" "; print $1}' |
61 awk '{print $1}'

62 # Retrieve the model
62 model=$(echo "$cacheentry" | /usr/xpg4/bin/awk '{FS=" "; print $2}')
64 model=$(echo "$cacheentry" | awk '{FS=" "; print $2}')

64 # Retrieve the driver
65 driver=$(echo "$cacheentry" | /usr/xpg4/bin/awk '{FS=" "; print $3}')
67 driver=$(echo "$cacheentry" | awk '{FS=" "; print $3}')

67 #
68 # Retrieve the PPD path. Parse the PPD path to get the
69 # label path and to figure out the repository letter
70 # associated with the label path. Note:
71 # the PPD file name is the 6th colon separated entry
72 # in the cache entry. This is may need to be modified if the
73 # format changes.
74 #
75 pppath=$(echo "$cacheentry" | /usr/xpg4/bin/awk '{FS=" "; print $6}')
77 pppath=$(echo "$cacheentry" | /bin/awk '{FS=" "; print $6}')
76 manupath=$(/bin/dirname "$pppath")
77 labelpath=$(/bin/dirname "$manupath")

79 case "$labelpath" in
80 /usr/share/ppd/*)
81     repltr=S
82     ;;
83 /opt/share/ppd/*)
84     repltr=V
85     ;;
86 /usr/local/share/ppd/*)
87     repltr=A
88     ;;
89 /var/lp/ppd/*)
90     repltr=U
91     ;;
92 esac

94 [[ -n "${repltr}" ]] || exit 1
95 echo "${manuf}\n${model}"
96 echo "$(/bin/basename "$labelpath")(${repltr}): $driver"

98 exit 0
```

new/usr/src/cmd/sendmail/cf/sh/check-permissions.sh

1

```
*****
3270 Sun May 26 18:22:46 2013
new/usr/src/cmd/sendmail/cf/sh/check-permissions.sh
XXX Remove awk(1)
*****
1 #!/bin/sh --
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Check :include: aliases (in files configured in sendmail.cf) and .forward
24 # files to make sure the files and their parent directory paths all have
25 # proper permissions. And check the master alias file(s) too.
26 #
27 # See http://www.sendmail.org/vendor/sun/migration.html#Security for details.
28 #
29 # Copyright 2006 Sun Microsystems, Inc. All rights reserved.
30 # Use is subject to license terms.
31 #
32 # %W% (Sun) %G%
33 # idont "%Z%M% %I% %E% SMI"
34 PATH=/bin
35 #
36 # Check the group- and world-writable bits on the given file.
37 #
38 analyze() {
39     case "`ls -l1dn $1`" in
40         ?????w??w?*)
41             echo $2: $1 is group and world writable
42             bogus_dirs=true ;;
43         ???????w?*)
44             echo $2: $1 is world writable
45             bogus_dirs=true ;;
46         ?????w????*)
47             echo $2: $1 is group writable
48             bogus_dirs=true ;;
49     esac
50 }
51 #
52 # unchanged_portion_omitted
53 #
54 #
55 #
56 #
57 config=/etc/mail/sendmail.cf
58 bogus_dirs=false
59 #
60 afl1=`grep "^OA" $config | sed 's/^OA//' | sed 's/,/ /g' | sed 's/.*/:/'`
61 afl2=`grep "^O AliasFile=" $config | sed 's/^O AliasFile=/' | \
62     sed 's/,/ /g' | sed 's/.*/:/'`
63 #
64 # These should be OK themselves, but other packages may have screwed up the
```

new/usr/src/cmd/sendmail/cf/sh/check-permissions.sh

2

```
85 # permissions on /etc or /etc/mail . And best to check in case non-standard
86 # alias paths are used.
87 #
88 break_down $afl1 $afl2
89 #
90 # Find all valid :include: files used in alias files configured in sendmail.cf
91 #
92 for i in `sed 's/^[#].*$//' $afl1 $afl2 | \
93     grep :include: | \
94     sed 's/.*:include:/' | \
95     sed 's/,.*$//'`
96 do
97     break_down $i
98 done
99 #
100 # Check .forward files as well. If the argument "ALL" is given, do it for
101 # everyone. If no argument to the script is given, just do it for the current
102 # user. O/w, do it for all arguments.
103 #
104 if [ $# -eq 0 ] ; then
105     arg='id | /usr/xpg4/bin/awk -F'(' '{n = split($2,id,")"; print id[1]}'`
106     arg='id | awk -F'(' '{n = split($2,id,")"; print id[1]}'`
107 elif [ $1 = "ALL" ] ; then
108     arg=""
109 else
110     arg="$*"
111 fi
112 #
113 for i in `getent passwd $arg | /usr/xpg4/bin/awk -F: '{print $6}`
114 for i in `getent passwd $arg | awk -F: '{print $6}`
115 do
116     if [ -f $i/.forward ] ; then
117         break_down $i/.forward
118     fi
119 done
120 #
121 $bogus_dirs || echo "No unsafe directories found."
```


new/usr/src/cmd/sendmail/lib/smtp-sendmail

1

```
*****
4093 Sun May 26 18:22:46 2013
new/usr/src/cmd/sendmail/lib/smtp-sendmail
XXX Remove awk(1)
*****
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.

25 . /lib/svc/share/smf_include.sh
26 . /lib/svc/share/sendmail_include.sh

28 ERRMSG1='WARNING: /var/mail is NFS-mounted without setting actimeo=0,'
29 ERRMSG2='this can cause mailbox locking and access problems.'
30 SERVER_PID_FILE="/var/run/sendmail.pid"
31 ALIASES_FILE="/etc/mail/aliases"
32 SENDMAIL_CF="/etc/mail/sendmail.cf"

34 case "$1" in
35 'refresh')
36     [ -f $SERVER_PID_FILE ] && kill -1 'head -1 $SERVER_PID_FILE'
37     ;;
38
39 'start')
40     exist_or_exit $SENDMAIL
41     if [ ! -d /var/spool/mqueue ]; then
42         /usr/bin/mkdir -m 0750 /var/spool/mqueue
43         /usr/bin/chown root:bin /var/spool/mqueue
44     fi
45     if [ ! -f $ALIASES_FILE.db ] && [ ! -f $ALIASES_FILE.dir ] \
46         && [ ! -f $ALIASES_FILE.pag ]; then
47         /usr/sbin/newaliases
48     fi
49     MODE="-bd"
50     [ -f $DEFAULT_FILE ] && . $DEFAULT_FILE
51     #
52     # * MODE should be "-bd" or null (MODE= or MODE="") or
53     # left alone. Anything else and you're on your own.
54     # * QUEUEOPTION should be "p" or null (as above).
55     # * QUEUEINTERVAL should be set to some legal value;
56     # sanity checks are done below.
57     # * OPTIONS are catch-alls; set with care.
58     #
59     if [ -n "$QUEUEOPTION" -a "$QUEUEOPTION" != "p" ]; then
60         QUEUEOPTION=""
61     fi
```

new/usr/src/cmd/sendmail/lib/smtp-sendmail

2

```
62     if [ -z "$QUEUEOPTION" -o -n "$QUEUEINTERVAL" ]; then
63         check_queue_interval_syntax $QUEUEINTERVAL
64         QUEUEINTERVAL=$answer
65     fi

67     local='/usr/bin/svcprop -p config/local_only $SMF_FMRI 2>/dev/null'
68     if [ $? -eq 0 -a "$local" = "true" ]; then
69         MODE="-bl"
70     fi
71     sendmail_path='svcprop -p config/path_to_sendmail_mc $SMF_FMRI \
72         2>/dev/null'
73     if [ $? -eq 0 -a -n "$sendmail_path" ]; then
74         turn_m4_crank "$SENDMAIL_CF" "$sendmail_path"
75     fi
76     exist_or_exit "$SENDMAIL_CF"

78     $SENDMAIL $MODE -q$QUEUEOPTION$QUEUEINTERVAL $OPTIONS &

80     #
81     # ETRN_HOSTS should be of the form
82     # "s1:c1.1,c1.2          s2:c2.1 s3:c3.1,c3.2,c3.3"
83     # i.e., white-space separated groups of server:client where
84     # client can be one or more comma-separated names; N.B. that
85     # the :client part is optional; see etrn(1M) for details.
86     # server is the name of the server to prod; a mail queue run
87     # is requested for each client name. This is comparable to
88     # running "/usr/lib/sendmail -qRclient" on the host server.
89     #
90     # See RFC 1985 for more information.
91     #
92     for i in $ETRN_HOSTS; do
93         SERVER='echo $i | /usr/bin/sed -e 's/:.*$//''
94         CLIENTS='echo $i | /usr/bin/sed -n -e 's/,/ /g' \
95             -e '/:s/^.*/:/'
96         /usr/sbin/etrn -b $SERVER $CLIENTS >/dev/null 2>&1 &
97     done

99     if /usr/xpg4/bin/awk 'BEGIN{s = 1}
99     if /usr/bin/nawk 'BEGIN{s = 1}
100         $2 == "/var/mail" && $3 == "nfs" && $4 !~ /actimeo=0/ &&
101         $4 !~ /noac/{s = 0} END{exit s}' /etc/mnttab; then

103         /usr/bin/logger -p mail.crit "$ERRMSG1"
104         /usr/bin/logger -p mail.crit "$ERRMSG2"
105     fi
106     ;;

108 'stop')
109     [ -f $SERVER_PID_FILE ] && check_and_kill $SERVER_PID_FILE
110     # Need to kill the entire service contract to kill all sendmail related
111     # processes
112     smf_kill_contract $2 TERM 1 30
113     ret=$?
114     [ $ret -eq 1 ] && exit 1

116     # Since sendmail spawns user processes out of .forward files, it is
117     # possible that some of these are not responding to TERM. If the
118     # contract did not empty after TERM, move on to KILL.
119     if [ $ret -eq 2 ]; then
120         smf_kill_contract $2 KILL 1
121     fi
122     ;;

124 *)
125     echo "Usage: $0 { start | stop | refresh }"
126     exit 1
```

`new/usr/src/cmd/sendmail/lib/smtp-sendmail`

3

```
127          ;;  
128 esac  
129 exit 0
```

new/usr/src/cmd/sgs/packages/Makefile.targ

1

1653 Sun May 26 18:22:47 2013

new/usr/src/cmd/sgs/packages/Makefile.targ

XXX Remove awk(1)

```
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.

24 pkginfo: ../common/pkginfo.tmpl awk_pkginfo
25 $(RM) $@; /usr/xpg4/bin/awk -f awk_pkginfo ../common/$@.tmpl > $@
25 $(RM) $@; nawk -f awk_pkginfo ../common/$@.tmpl > $@

27 pkg: FRC
28 @ $(RM) -r $(PACKAGE)
29 pkgmk -f prototype_$(MACH) -d $(PKGARCHIVE) -r $(ROOT) \
30 -o $(PACKAGE) 2>&1 | egrep -v "$(SRC)|parametric paths may"
31 pkgtrans -o -s $(PKGARCHIVE) $(PACKAGE).tmp $(PACKAGE)
32 $(RM) -r $(PKGARCHIVE)/$(PACKAGE)
33 mv $(PKGARCHIVE)/$(PACKAGE).tmp $(PKGARCHIVE)/$(PACKAGE)

35 $(DATAFILES): ../common/$$@
36 $(RM) $@; cp ../common/$@ $@

38 awk_pkginfo: ../bld_awk_pkginfo ../common/readme_revision ../common/$(README)
39 ../bld_awk_pkginfo \
40 -R `perl ../common/readme_revision -d ../common/$(README)` \
41 -m $(MACH) -r "$(RELEASE)" -o $@

43 clean: FRC
44 $(RM) $(CLEANFILES)

46 clobber: clean FRC
47 $(RM) -r $(CLOBBERFILES)

49 FRC:
```

new/usr/src/cmd/sgs/tools/chkmsg.sh

1

```
*****
3221 Sun May 26 18:22:47 2013
new/usr/src/cmd/sgs/tools/chkmsg.sh
XXX Remove awk(1)
*****
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 #
24 # Copyright 2006 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 # ident "%Z%M% %I% %E% SMI"

28 TOOLDIR="{SRC}/cmd/sgs/tools/"

30 #
31 # remove the temporary files
32 #
33 rm -f CATA_MSG_INTL_LIST CATA_MSG_ORIG_LIST
34 rm -f MSG_INTL_LIST MSG_ORIG_LIST

36 while getopts "m:" Arg
37 do
38     case $Arg in
39         m) /usr/xpg4/bin/awk -f ${TOOLDIR}/catalog.awk $OPTARG ;;
40         m) awk -f ${TOOLDIR}/catalog.awk $OPTARG ;;
41         \?) echo "usage: chkmsg -m msgfile source-files" ; exit 1 ;;
42         esac
43 done
44 shift `expr $OPTIND - 1`

45 if [ $# -eq 0 ]; then
46     echo "usage: chkmsg -m msgfile source-files"
47     exit 1
48 fi

50 #
51 # Sort the MSG_INTL() and MSG_ORIG() entries. Note, messages can come in _32
52 # and _64 flavors - if so strip the suffix and uniquify the output.
53 #
54 if [ -s CATA_MSG_INTL_LIST ]; then
55     sed -e "s/_32$//" -e "s/_64$//" CATA_MSG_INTL_LIST | sort | uniq > _TMP
56     mv _TMP CATA_MSG_INTL_LIST
57 fi
58 if [ -s CATA_MSG_ORIG_LIST ]; then
59     sed -e "s/_32$//" -e "s/_64$//" CATA_MSG_ORIG_LIST | sort | uniq > _TMP
```

new/usr/src/cmd/sgs/tools/chkmsg.sh

2

```
60     mv _TMP CATA_MSG_ORIG_LIST
61 fi

63 #
64 # Generate the lists for the source files and sort them
65 #
66 /usr/xpg4/bin/awk -f ${TOOLDIR}/getmessage.awk $*
67 awk -f ${TOOLDIR}/getmessage.awk $*

68 if [ -s MSG_INTL_LIST ]; then
69     sed -e "s/_32$//" -e "s/_64$//" MSG_INTL_LIST | sort | uniq > _TMP
70     mv _TMP MSG_INTL_LIST
71 fi
72 if [ -s MSG_ORIG_LIST ]; then
73     sed -e "s/_32$//" -e "s/_64$//" MSG_ORIG_LIST | sort | uniq > _TMP
74     mv _TMP MSG_ORIG_LIST
75 fi

77 #
78 # Start checking
79 #
80 Error=0

82 #
83 # Check MSG_INTL message
84 #
85 comm -23 CATA_MSG_INTL_LIST MSG_INTL_LIST > _TMP 2> /dev/null
86 if [ -s _TMP ]; then
87     echo
88     echo "messages exist between _START_ and _END_ but do not use MSG_INTL()"
89     cat _TMP | sed "s/^/"
90     Error=1
91 fi
92 rm -f _TMP

94 comm -13 CATA_MSG_INTL_LIST MSG_INTL_LIST > _TMP 2> /dev/null
95 if [ -s _TMP ]; then
96     echo
97     echo "use of MSG_INTL() but messages do not exist between _START_ and _END_"
98     cat _TMP | sed "s/^/"
99     Error=1
100 fi
101 rm -f _TMP

103 #
104 # Check MSG_ORIG message
105 #
106 comm -23 CATA_MSG_ORIG_LIST MSG_ORIG_LIST > _TMP 2> /dev/null
107 if [ -s _TMP ]; then
108     echo
109     echo "messages exist after _END_ but do not use MSG_ORIG()"
110     cat _TMP | sed "s/^/"
111     Error=1
112 fi
113 rm -f _TMP

115 comm -13 CATA_MSG_ORIG_LIST MSG_ORIG_LIST > _TMP 2> /dev/null
116 if [ -s _TMP ]; then
117     echo
118     echo "use of MSG_ORIG() but messages do not exist after _END_"
119     cat _TMP | sed "s/^/"
120     Error=1
121 fi
122 rm -f _TMP

124 #
```

new/usr/src/cmd/sgs/tools/chkmsg.sh

3

```
125 # remove the temporary files
126 #
127 rm -f CATA_MSG_INTL_LIST CATA_MSG_ORIG_LIST
128 rm -f MSG_INTL_LIST MSG_ORIG_LIST

130 exit $Error
```

```

*****
3363 Sun May 26 18:22:48 2013
new/usr/src/cmd/svc/milestone/fs-local
XXX Remove nawk(1)
*****
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright (c) 1988, 2010, Oracle and/or its affiliates. All rights reserved.
24 #

26 . /lib/svc/share/smf_include.sh

28 result=$SMF_EXIT_OK
29 mntretry=0

31 # Mount all local filesystems.

33 cd /; /sbin/mountall -l >/var/run/fs-local 2>&1
34 rc=$?
35 if [ $rc -eq 111 ]; then
36 #
37 # The only failures were lofs mounts, we can try again
38 # after zfs is mounted, there is a chance a lofs mount
39 # failed due to it depending on a zfs not yet mounted.
40 #
41 mntretry=1
42 elif [ $rc -ne 0 ]; then
43 cat /var/run/fs-local >/dev/msglog
44 msg="WARNING: /sbin/mountall -l failed: exit status $rc"
45 echo $msg
46 echo "$SMF_FMRI:" $msg >/dev/msglog
47 result=$SMF_EXIT_ERR_FATAL
48 fi
49 rm -f /var/run/fs-local

51 #
52 # If there are non-global UFS filesystems with quotas, check and enable them.
53 #

55 # vlist is the non-global filesystems in vfstab requesting quotas
56 vlist='/usr/xpg4/bin/awk '$1 !~ /^(#|-)/ && $4 == "ufs" {
57 vlist='/usr/bin/nawk '$1 !~ /^(#|-)/ && $4 == "ufs" {
58 if (match($7, "(^|,)(quota|rq)(,|$)" != 0 &&
match($7, "(^|,)(global(,|$)" == 0) print $1; }' /etc/vfstab'

60 if [ -n "$vlist" ]; then

```

```

61 # mlist is the filesystems in mnttab that are ufs, mounted rw,
62 # and without quotas turned on
63 mlist='/usr/sbin/mount -p | /usr/xpg4/bin/awk '$4 == "ufs" {
63 mlist='/usr/sbin/mount -p | /usr/bin/nawk '$4 == "ufs" {
64 if (match($7, "(^|,)(ro(,|$)" == 0) print $1; }'

66 # qlist is the intersection of vlist and mlist
67 qlist='echo "$vlist\n-\n$mlist" | \
68 /usr/xpg4/bin/awk '{if ($1 == "-") { mlist = 1; }
68 /usr/bin/nawk '{if ($1 == "-") { mlist = 1; }
69 else if (mlist == 0) { vlist[$1] = 1; }
70 else if (vlist[$1]) { print $1; } }'

72 #
73 # Just check and enable the non-global UFS file systems with quotas
74 # enabled. Note that "quotacheck -a" and "quotaon -a" will try
75 # to process all UFS entries with quotas rather than excluding
76 # the entries with the global option (the global entries are handled
77 # later in another script if the cluster package is installed).
78 #
79 if [ -n "$qlist" ]; then
80 echo 'Checking UFS quotas: \c'
81 /usr/sbin/quotacheck -p $qlist
82 echo 'done.'
83 /usr/sbin/quotaon $qlist
84 fi
85 fi

87 # Mount all ZFS filesystems.

89 if [ -x /usr/sbin/zfs ]; then
90 /usr/sbin/zfs mount -va >/dev/msglog 2>&1
91 rc=$?
92 if [ $rc -ne 0 ]; then
93 msg="WARNING: /usr/sbin/zfs mount -a failed: exit status $rc"
94 echo $msg
95 echo "$SMF_FMRI:" $msg >/dev/msglog
96 result=$SMF_EXIT_ERR_FATAL
97 fi
98 fi

100 if [ $result = $SMF_EXIT_OK -a $mntretry -eq 1 ]
101 then
102 cd /; /sbin/mountall -l >/dev/msglog
103 rc=$?
104 if [ $rc -ne 0 ]; then
105 msg="WARNING: /sbin/mountall -l failed: exit status $rc"
106 echo $msg
107 echo "$SMF_FMRI:" $msg >/dev/msglog
108 result=$SMF_EXIT_ERR_FATAL
109 fi
110 fi

112 # Add swap filesystems
113 /sbin/swapadd >/dev/null 2>&1

115 exit $result

```

new/usr/src/cmd/svc/milestone/fs-usr

1

```
*****
5698 Sun May 26 18:22:48 2013
new/usr/src/cmd/svc/milestone/fs-usr
XXX Remove awk(1)
*****
_unchanged_portion_omitted_

63 #
64 # Write a unique id into this kernel image; this will be included
65 # in the dump header and panicbuf of any crashdump of this image.
66 #
67 if [ -x /usr/sbin/dumpadm ]; then
68     /usr/sbin/dumpadm -i
69 fi

71 rootiszfs=0
72 # get the fstype of root
73 readmnttab / </etc/mnttab
74 if [ "$fstype" = zfs ]; then
75     rootiszfs=1
76     dump_setup
77 fi

79 #
80 # Add physical swap.
81 #
82 /sbin/swapadd -l

84 #
85 # Check and remount the / (root) file system.
86 # For NFS mounts, force the llock option on.
87 #
88 if smf_is_globalzone && [ $rootiszfs = 0 ]; then
89     readvfstab / < $vfstab
90     checkfs $fsckdev $fstype $mountp || exit $SMF_EXIT_ERR_FATAL
91     checkopt "llock" $mntopts
92     mntopts='remount'

94     [ -n "$otherops" ] && mntopts="${mntopts},${otherops}"
95     [ "$fstype" = nfs ] && mntopts="${mntopts},llock"

97     # if root dev is a read-only metadvice then fail
98     case $special in
99         /dev/md/dsk/*)
100             dd if=/dev/null of=$special count=0 >/dev/null 2>&1 ||
101                 exit $SMF_EXIT_ERR_FATAL
102             ;;
103         esac

105     mountfs -m $mountp $fstype $mntopts - || exit $SMF_EXIT_ERR_FATAL
106 fi

108 #
109 # Check and remount the /usr file system (formerly mounted read-only).
110 # Unless root is zfs, in which case we've already mounted /usr read-write
111 #
112 if [ "$rootiszfs" = 0 ]; then
113     readvfstab /usr < $vfstab
114     if [ "$mountp" ]; then
115         if [ "$fstype" = cachefs ]; then
116             mountfs -O $mountp cachefs $mntopts $special ||
117                 exit $SMF_EXIT_ERR_FATAL
118         else
119             checkopt ro $mntopts
120             if [ "$x$option" != xro ]; then
121                 checkfs $fsckdev $fstype $mountp ||
```

new/usr/src/cmd/svc/milestone/fs-usr

2

```
122         exit $SMF_EXIT_ERR_FATAL
123     if [ "$mntopts" != x- ]; then
124         mntopts="remount,$mntopts"
125     else
126         mntopts="remount"
127     fi

129     # if usr dev is a read-only metadvice then fail
130     case $special in
131         /dev/md/dsk/*)
132             dd if=/dev/null of=$special count=0 \
133                 >/dev/null 2>&1 || exit $SMF_EXIT_ER
134             ;;
135         esac

137     mountfs - /usr $fstype $mntopts - ||
138         exit $SMF_EXIT_ERR_FATAL
139     fi
140     fi
141     fi
142 fi

144 #
145 # Check and mount the /usr/platform file system. This should only be
146 # present when a SunOS 5.5 (Solaris 2.5) or greater client is being
147 # administered by a SunOS 5.4 or less host.
148 #
149 readvfstab /usr/platform < $vfstab
150 if [ "$mountp" ]; then
151     checkfs $fsckdev $fstype $mountp || exit $SMF_EXIT_ERR_FATAL
152     mountfs - $mountp $fstype $mntopts - || exit $SMF_EXIT_ERR_FATAL
153 fi

155 #
156 # Mount the fd file systems if mount point exists.
157 #
158 readvfstab /dev/fd < $vfstab
159 if [ "$mountp" -a -d /dev/fd ]; then
160     mountfs - /dev/fd - - - || exit $SMF_EXIT_ERR_FATAL
161 fi

163 if [ -f "${UPDATEFILE}" ]; then
164     /usr/sbin/bootadm update-archive
165     if [ $? != 0 ]; then
166         cecho ""
167         cecho "WARNING: Automatic update of the boot archive failed."
168         cecho "Update the archives using 'bootadm update-archive'"
169         cecho "command and then reboot the system from the same device t
170         cecho "was previously booted."
171         cecho ""
172         exit $SMF_EXIT_ERR_FATAL
173     fi
174     rm -f $UPDATEFILE
175     bootcmd="/usr/sbin/eeprom bootcmd | /usr/bin/sed -e 's#bootcmd=##g'"
176     if [ 'uname -p' = "i386" ]; then
177         /usr/sbin/reboot -f dryrun
178         if [ $? = 0 ]; then
179             /usr/sbin/reboot -f -- "$bootcmd"
180             exit $SMF_EXIT_OK
181         fi
182     boot_prop="/usr/sbin/svcconf -s svc:/system/boot-config:default l
183         /usr/xpg4/bin/awk '{ print $3}'"
184         /usr/bin/nawk '{ print $3}'"
185     if [ "$boot_prop" != "true" ]; then
186         cecho ""
187         cecho "WARNING: Reboot required."
```

```
187         cecho "The system has updated the cache of files (boot a
188         cecho "during the early boot sequence. To avoid booting
189         cecho "with the previously out-of-sync version of these
190         cecho "system from the same device that was previously b
191         cecho ""
192         exit $SMF_EXIT_ERR_FATAL
193     else
194         /usr/sbin/reboot -p
195         exit $SMF_EXIT_OK
196     fi
197 fi
198 /usr/sbin/reboot -- "$bootcmd"
199 fi

201 exit $SMF_EXIT_OK
```


new/usr/src/cmd/svc/milestone/net-loc

1

```
*****
18652 Sun May 26 18:22:49 2013
new/usr/src/cmd/svc/milestone/net-loc
XXX Remove awk(1)
*****
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License at the following language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25
26 . /lib/svc/share/smf_include.sh
27 . /lib/svc/share/net_include.sh
28
29 # FMRI consts
30 AUTOFS_FMRI="svc:/system/filesystem/autofs"
31 DNS_CLIENT_FMRI="svc:/network/dns/client"
32 IPSEC_IKE_FMRI="svc:/network/ipsec/ike"
33 IPSEC_POLICY_FMRI="svc:/network/ipsec/policy"
34 IPFILTER_FMRI="svc:/network/ipfilter:default"
35 LDAP_CLIENT_FMRI="svc:/network/ldap/client"
36 LOCATION_FMRI="svc:/network/location:default"
37 MAPID_FMRI="svc:/network/nfs/mapid:default"
38 NIS_CLIENT_FMRI="svc:/network/nis/client"
39 NWAM_FMRI="svc:/network/physical:nwam"
40
41 # commands
42 CP=/usr/bin/cp
43 DHCPINFO=/sbin/dhcpinfo
44 DOMAINNAME=/usr/bin/domainname
45 GREP=/usr/bin/grep
46 LDAPCLIENT=/usr/sbin/ldapclient
47 MKDIR=/usr/bin/mkdir
48 MV=/usr/bin/mv
49 AWK=/usr/xpg4/bin/awk
50 NAWK=/usr/bin/nawk
51 NWAMADM=/usr/sbin/nwamadm
52 NWAMCFG=/usr/sbin/nwamcfg
53 RM=/usr/bin/rm
54 SED=/usr/bin/sed
55 SVCADM=/usr/sbin/svcadm
56 SVCCFG=/usr/sbin/svccfg
57 SVCPROP=/usr/bin/svcpop
58 TOUCH=/usr/bin/touch
59 # Path to directories
60 ETC_DEFAULT_DOMAIN=/etc/defaultdomain
```

new/usr/src/cmd/svc/milestone/net-loc

2

```
61 NIS_BIND_PATH=/var/yp/binding
62 LEGACY_LOC_PATH=/etc/nwam/loc/Legacy
63 USER_LOC_PATH=/etc/nwam/loc/User
64 SCRIPT_PATH=/etc/svc/volatile/nwam
65
66 #
67 # echoes DHCP controlled interfaces separated by commas
68 #
69 # Don't parse the output of ifconfig(1M) because interfaces that haven't
70 # acquired a DHCP lease also have the DHCP flag set.
71 #
72 get_dhcp_interfaces () {
73 #
74 # 1. parse netstat(1M) output for v4 interfaces in BOUND
75 # or INFORMATION state
76 # 2. make a space-separated list of interface names
77 #
78 netstat -D -f inet | $AWK '
78 netstat -D -f inet | $NAWK '
79 $2 ~ /BOUND/ { printf "%s ", $1 }
80 $2 ~ /INFORMATION/ { printf "%s ", $1 }'
81 }
unchanged portion omitted
82
170 #
171 # do_dns <location>
172 #
173 # Installs DNS information on /etc/resolv.conf for location
174 #
175 # Returns 0 on success, 1 on failure
176 #
177 do_dns () {
178 loc=$1
179 file=/etc/resolv.conf
180
181 # Write out to temporary file first
182 $TOUCH $file.$$
183
184 DNS_CONFIGSRC='nwam_get_loc_list_prop $loc dns-nameservice-configsrc'
185 if [ -z "$DNS_CONFIGSRC" ]; then
186 echo "missing 'dns-nameservice-configsrc' property for '$loc'"
187 return 1
188 fi
189
190 for configsrc in $DNS_CONFIGSRC; do
191 case "$configsrc" in
192 'manual')
193 DNS_SERVERS='nwam_get_loc_list_prop $loc \
194 dns-nameservice-servers'
195 if [ -z "$DNS_SERVERS" ]; then
196 echo "DNS nameserver not set for '$loc'"
197 return 1
198 fi
199 DNS_DOMAIN='nwam_get_loc_prop $loc \
200 dns-nameservice-domain'
201 DNS_SEARCH='nwam_get_loc_list_prop $loc \
202 dns-nameservice-search'
203 ;;
204 'dhcp')
205 DNS_SEARCH='get_dhcpinfo DNSdomain'
206 DNS_SERVERS='get_dhcpinfo DNSserv'
207 # Use first search list entry as default domain
208 set -- $DNS_SEARCH
209 DNS_DOMAIN=$1
210 ;;
211 '**')

```

```

212         echo "Unrecognized DNS configsrc ${configsrc}; ignoring"
213         ;;
214     esac

216     # Write DNS settings
217     if [ -n "$DNS_DOMAIN" ]; then
218         echo "$DNS_DOMAIN" | $AWK \
218         echo "$DNS_DOMAIN" | $NAWK \
219         '{ for (i = 1; i <= NF; i++) \
220         print "domain ", $i }' >> $file.$$
221     fi
222     if [ -n "$DNS_SEARCH" ]; then
223         echo "$DNS_SEARCH" | $AWK \
223         echo "$DNS_SEARCH" | $NAWK \
224         '{ printf("search"); \
225         for (i = 1; i <= NF; i++) printf(" %s", $i); \
226         printf("\n") }' >> $file.$$
227     fi
228     if [ -n "$DNS_SERVERS" ]; then
229         echo "$DNS_SERVERS" | $AWK \
229         echo "$DNS_SERVERS" | $NAWK \
230         '{ for (i = 1; i <= NF; i++) \
231         print "nameserver ", $i }' >> $file.$$
232     fi
233 done

235 # Finally, copy our working version to the real thing
236 $MV -f $file.$$ $file
237 start_svc $DNS_CLIENT_FMRI

239 return 0
240 }

242 #
243 # do_nis <location>
244 #
245 # Installs NIS information on /var/yp/binding/ for location
246 #
247 # Returns 0 on success, 1 on failure
248 #
249 do_nis () {
250     loc=$1

252     NIS_CONFIGSRC='nwam_get_loc_list_prop $loc nis-nameservice-configsrc'
253     if [ -z "$NIS_CONFIGSRC" ]; then
254         echo "missing 'nis-nameservice-configsrc' property for '$loc'"
255         return 1
256     fi

258     for configsrc in $NIS_CONFIGSRC; do
259         case "$configsrc" in
260             'manual')
261                 NIS_SERVERS='nwam_get_loc_list_prop $loc \
262                 nis-nameservice-servers'
263                 DEFAULT_DOMAIN='nwam_get_loc_prop $loc default-domain'
264                 # user-specified default-domain always wins
265                 if [ -n "$DEFAULT_DOMAIN" ]; then
266                     $DOMAINNAME $DEFAULT_DOMAIN
267                     $DOMAINNAME > $ETC_DEFAULT_DOMAIN
268                 else
269                     echo "'domainname' not set for '$loc'"
270                     return 1
271                 fi
272             ;;
273             'dhcp')
274                 # Use only the first name

```

```

275     DEFAULT_DOMAIN='get_dhcpinfo NISdmain | \
276     $AWK '{ print $1 }' \
276     $NAWK '{ print $1 }' \
277     NIS_SERVERS='get_dhcpinfo NISservs'
278     $DOMAINNAME $DEFAULT_DOMAIN
279     $DOMAINNAME > $ETC_DEFAULT_DOMAIN
280     ;;
281     '**')
282         echo "Unrecognized NIS configsrc ${configsrc}; ignoring"
283         ;;
284     esac

286     # Place NIS settings in appropriate directory/file.
287     if [ ! -d "$NIS_BIND_PATH/$DEFAULT_DOMAIN" ]; then
288         $MKDIR -p $NIS_BIND_PATH/$DEFAULT_DOMAIN
289     fi
290     if [ -n "$NIS_SERVERS" ]; then
291         echo "$NIS_SERVERS" | $AWK \
291         echo "$NIS_SERVERS" | $NAWK \
292         '{ for (i = 1; i <= NF; i++) print $i }' \
293         > $NIS_BIND_PATH/$DEFAULT_DOMAIN/ypservers
294     fi
295 done

297 start_svc $NIS_CLIENT_FMRI

299 return 0
300 }
unchanged_portion_omitted

```

new/usr/src/cmd/svc/milestone/net-nwam

1

```
*****
19170 Sun May 26 18:22:49 2013
new/usr/src/cmd/svc/milestone/net-nwam
XXX Remove nawk(1)
*****
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright (c) 2007, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright 2012 Milan Jurik. All rights reserved.
25 #
26
27 . /lib/svc/share/smf_include.sh
28 . /lib/svc/share/net_include.sh
29
30 # FMRI constants
31 IPSEC_IKE_FMRI="svc:/network/ipsec/ike"
32 IPSEC_POLICY_FMRI="svc:/network/ipsec/policy"
33 IPFILTER_FMRI="svc:/network/ipfilter:default"
34 NIS_CLIENT_FMRI="svc:/network/nis/client:default"
35 NET_PHYS_FMRI="svc:/network/physical:default"
36 NET_NWAM_FMRI="svc:/network/physical:nwam"
37 NET_LOC_FMRI="svc:/network/location:default"
38 NFS_MAPID_FMRI="svc:/network/nfs/mapid:default"
39
40 #
41 # Default *.conf files
42 # Set appropriate config SMF property to these files when NWAM is stopped
43 # and corresponding config properties in the Legacy location are empty
44 #
45 IPF6_DEFAULT_CONFIG_FILE=/etc/ipf/ipf6.conf
46 IPNAT_DEFAULT_CONFIG_FILE=/etc/ipf/ipnat.conf
47 IPPPOOL_DEFAULT_CONFIG_FILE=/etc/ipf/ippool.conf
48 IPSEC_IKE_DEFAULT_CONFIG_FILE=/etc/inet/ike/config
49 IPSEC_POLICY_DEFAULT_CONFIG_FILE=/etc/inet/ipsecinit.conf
50
51 # commands
52 BASENAME=/usr/bin/basename
53 CAT=/usr/bin/cat
54 CP=/usr/bin/cp
55 DOMAINNAME=/usr/bin/domainname
56 GREP=/usr/bin/grep
57 LDAPCLIENT=/usr/sbin/ldapclient
58 MKDIR=/usr/bin/mkdir
59 MKFIFO=/usr/bin/mkfifo
60 AWK=/usr/xpg4/bin/awk
61 NWAWK=/usr/bin/nawk
```

new/usr/src/cmd/svc/milestone/net-nwam

2

```
61 NWAMCFG=/usr/sbin/nwamcfg
62 RM=/usr/bin/rm
63 SVCADM=/usr/sbin/svcadm
64 SVCCFG=/usr/sbin/svccfg
65 SVCPROP=/usr/bin/svcprop
66
67 # Path to directories
68 # We don't have a writable file system so we write to /etc/svc/volatile and
69 # then later copy anything interesting to /etc/nwam.
70 VOL_NWAM_PATH=/etc/svc/volatile/nwam
71 VOL_LEGACY_PATH=$VOL_NWAM_PATH/Legacy
72 PERM_LEGACY_PATH=/etc/nwam/loc/Legacy
73 NIS_BIND_PATH=/var/yp/binding
74
75 #
76 # copy_to_legacy_loc <file>
77 #
78 # Copies the file to the Legacy location directory
79 # (in /etc/svc/volatile/nwam/Legacy)
80 #
81 copy_to_legacy_loc() {
82     $MKDIR -p $VOL_LEGACY_PATH
83     if [ -f "$1" ]; then
84         $CP -p $1 $VOL_LEGACY_PATH
85     fi
86 }
87
88 unchanged_portion_omitted
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117 #
118 # Creates Legacy location from the current configuration
119 #
120 #
121 create_legacy_loc () {
122     CREATE_LOC_LEGACY_FILE=$VOL_NWAM_PATH/create_loc_legacy
123
124     #
125     # Write nwamcfg commands to create Legacy location to
126     # $CREATE_LOC_LEGACY_FILE as values for properties are determined
127     # Note that some of the *_CONFIG_FILE variables point at copies of
128     # files we've made and others indicate where those copies should be
129     # if we are enabling the location.
130     #
131     echo "create loc Legacy" > $CREATE_LOC_LEGACY_FILE
132     write_loc_prop "activation-mode" "system" $CREATE_LOC_LEGACY_FILE
133
134     NAMESERVICES=""
135     NAMESERVICES_CONFIG_FILE=""
136     DNS_NAMESERVICE_CONFIGSRC=""
137     DNS_NAMESERVICE_DOMAIN=""
138     DNS_NAMESERVICE_SERVERS=""
139     DNS_NAMESERVICE_SEARCH=""
140     NIS_NAMESERVICE_CONFIGSRC=""
141     NIS_NAMESERVICE_SERVERS=""
142     LDAP_NAMESERVICE_CONFIGSRC=""
143     LDAP_NAMESERVICE_SERVERS=""
144     DEFAULT_DOMAIN=""
145
146     # Copy /etc/nsswitch.conf file
147     copy_to_legacy_loc /etc/nsswitch.conf
148     NAMESERVICES_CONFIG_FILE="$VOL_LEGACY_PATH/nsswitch.conf"
149
150     # Gather DNS info from resolv.conf if present.
151     if [ -f /etc/resolv.conf ]; then
152         NAMESERVICES="dns,"
153         $GREP -i "added by dhcp" /etc/nsswitch.conf >/dev/null
154         if [ $? -eq 0 ]; then
155             DNS_NAMESERVICE_CONFIGSRC="dhcp"
156         fi
157     fi
158 }
```

```

175     else
176         DNS_NAMESERVICE_CONFIGSRC="manual"
177         DNS_NAMESERVICE_DOMAIN='${SAWK ' $1 == "domain" { \
177         DNS_NAMESERVICE_DOMAIN=' $NAWK ' $1 == "domain" { \
178             print $2 }' < /etc/resolv.conf'
179         DNS_NAMESERVICE_SERVERS='${SAWK ' $1 == "nameserver" \
179         DNS_NAMESERVICE_SERVERS=' $NAWK ' $1 == "nameserver" \
180             { printf "%s," $2 }' < /etc/resolv.conf'
181         DNS_NAMESERVICE_SEARCH='${SAWK ' $1 == "search" \
181         DNS_NAMESERVICE_SEARCH=' $NAWK ' $1 == "search" \
182             { printf "%s," $2 }' < /etc/resolv.conf'
183         copy_to_legacy_loc /etc/resolv.conf
184     fi
185 fi

187 # Gather NIS info from appropriate file if present.
188 if service_is_enabled $NIS_CLIENT_FMRI; then
189     NAMESERVICES="{NAMESERVICES}nis,"
190     NIS_NAMESERVICE_CONFIGSRC="manual"
191     DEFAULT_DOMAIN='${CAT /etc/defaultdomain}'

193     yp_servers='${SAWK '{ printf "%s " $1 }' \
193     yp_servers='${NAWK '{ printf "%s " $1 }' \
194     < $NIS_BIND_PATH/$DEFAULT_DOMAIN/ypservers'
195     for serv in $yp_servers; do
196         if is_valid_addr $serv; then
197             addr="$serv,"
198         else
199             addr='$(GREP -iw $serv /etc/inet/hosts | \
200                 $SAWK '{ printf "%s," $1 }' \
200                 $NAWK '{ printf "%s," $1 }'
201         fi
202         NIS_NAMESERVICE_SERVERS="{NIS_NAMESERVICE_SERVERS}$addr
203     done
204 fi

206 # Gather LDAP info via ldapclient(1M).
207 if [ -f /var/ldap/ldap_client_file ]; then
208     copy_to_legacy /var/ldap/ldap_client_file
209     NAMESERVICES="{NAMESERVICES}ldap,"
210     LDAP_NAMESERVICE_CONFIGSRC="manual"
211     LDAP_NAMESERVICE_SERVERS='$(LDAPCLIENT list 2>/dev/null | \
212         $SAWK ' $1 == "preferredServerList:" { print $2 }' \
212         $NAWK ' $1 == "preferredServerList:" { print $2 }'
213     DEFAULT_DOMAIN='${CAT /etc/defaultdomain}'
214 fi

216 # Now, write nwamcfg commands for nameservices
217 write_loc_prop "nameservices" $NAMESERVICES $CREATE_LOC_LEGACY_FILE
218 write_loc_prop "nameservices-config-file" $NAMESERVICES_CONFIG_FILE \
219     $CREATE_LOC_LEGACY_FILE
220 write_loc_prop "dns-nameservice-configsrc" $DNS_NAMESERVICE_CONFIGSRC \
221     $CREATE_LOC_LEGACY_FILE
222 write_loc_prop "dns-nameservice-domain" $DNS_NAMESERVICE_DOMAIN \
223     $CREATE_LOC_LEGACY_FILE
224 write_loc_prop "dns-nameservice-servers" $DNS_NAMESERVICE_SERVERS \
225     $CREATE_LOC_LEGACY_FILE
226 write_loc_prop "dns-nameservice-search" $DNS_NAMESERVICE_SEARCH \
227     $CREATE_LOC_LEGACY_FILE
228 write_loc_prop "nis-nameservice-configsrc" $NIS_NAMESERVICE_CONFIGSRC \
229     $CREATE_LOC_LEGACY_FILE
230 write_loc_prop "nis-nameservice-servers" $NIS_NAMESERVICE_SERVERS \
231     $CREATE_LOC_LEGACY_FILE
232 write_loc_prop "ldap-nameservice-configsrc" $LDAP_NAMESERVICE_CONFIGSRC \
233     $CREATE_LOC_LEGACY_FILE
234 write_loc_prop "ldap-nameservice-servers" $LDAP_NAMESERVICE_SERVERS \

```

```

235     $CREATE_LOC_LEGACY_FILE
236 write_loc_prop "default-domain" $DEFAULT_DOMAIN $CREATE_LOC_LEGACY_FILE

238 # Retrieve NFSv4 domain from SMF.
239 if service_is_enabled $NFS_MAPID_FMRI; then
240     NFS_DOMAIN='get_smf_prop NFS_MAPID_FMRI \
241         nfs-props/nfsmapid_domain'
242     write_loc_prop "nfsv4-domain" \
243         $NFS_DOMAIN $CREATE_LOC_LEGACY_FILE
244 fi

246 IPF_CONFIG_FILE=""
247 IPF6_CONFIG_FILE=""
248 IPNAT_CONFIG_FILE=""
249 IPPPOOL_CONFIG_FILE=""
250 IKE_CONFIG_FILE=""
251 IPSEC_POLICY_CONFIG_FILE=""

253 #
254 # IPFilter
255 #
256 # If the firewall policy is "custom", simply copy the
257 # custom_policy_file. If the firewall policy is "none", "allow" or
258 # "deny", save the value as "/<value>". When reverting back to the
259 # Legacy location, these values will have to be treated as special.
260 #
261 # For all configuration files, copy them to the Legacy directory.
262 # Use the respective properties to remember the original locations
263 # of the files so that they can be copied back there when NWAM is
264 # stopped.
265 #
266 if service_is_enabled $IPFILTER_FMRI; then
267     FIREWALL_POLICY='get_smf_prop $IPFILTER_FMRI \
268         firewall_config_default/policy'
269     if [ "$FIREWALL_POLICY" = "custom" ]; then
270         IPF_CONFIG_FILE='get_smf_prop $IPFILTER_FMRI \
271             firewall_config_default/custom_policy_file'
272         copy_to_legacy_loc $IPF_CONFIG_FILE
273     else
274         # save value as /none, /allow, or /deny
275         IPF_CONFIG_FILE="/$FIREWALL_POLICY"
276     fi
277     IPF6_CONFIG_FILE='get_smf_prop $IPFILTER_FMRI \
278         config/ipf6_config_file'
279     copy_to_legacy_loc $IPF6_CONFIG_FILE

281     IPNAT_CONFIG_FILE='get_smf_prop $IPFILTER_FMRI \
282         config/ipnat_config_file'
283     copy_to_legacy_loc $IPNAT_CONFIG_FILE

285     IPPPOOL_CONFIG_FILE='get_smf_prop $IPFILTER_FMRI \
286         config/ippool_config_file'
287     copy_to_legacy_loc $IPPPOOL_CONFIG_FILE
288 fi

290 # IKE
291 if service_is_enabled $IPSEC_IKE_FMRI:default; then
292     IKE_CONFIG_FILE='get_smf_prop $IPSEC_IKE_FMRI config/config_file'
293     copy_to_legacy_loc $IKE_CONFIG_FILE
294 fi

296 # IPsec
297 if service_is_enabled $IPSEC_POLICY_FMRI:default; then
298     IPSEC_POLICY_CONFIG_FILE='get_smf_prop $IPSEC_POLICY_FMRI \
299         config/config_file'
300     copy_to_legacy_loc $IPSEC_POLICY_CONFIG_FILE

```

```

301     fi
302
303     if [ -n "$IPF_CONFIG_FILE" -a \( "$IPF_CONFIG_FILE" = "/allow" \
304         -o "$IPF_CONFIG_FILE" = "/deny" -o "$IPF_CONFIG_FILE" = "/none" \
305         -o -f "$IPF_CONFIG_FILE" \) ]; then
306         write_loc_prop "ipfilter-config-file" $IPF_CONFIG_FILE \
307             $CREATE_LOC_LEGACY_FILE
308     fi
309     if [ -n "$IPF6_CONFIG_FILE" -a -f "$IPF6_CONFIG_FILE" ]; then
310         write_loc_prop "ipfilter-v6-config-file" $IPF6_CONFIG_FILE \
311             $CREATE_LOC_LEGACY_FILE
312     fi
313     if [ -n "$IPNAT_CONFIG_FILE" -a -f "$IPNAT_CONFIG_FILE" ]; then
314         write_loc_prop "ipnat-config-file" $IPNAT_CONFIG_FILE \
315             $CREATE_LOC_LEGACY_FILE
316     fi
317     if [ -n "$IPPOOL_CONFIG_FILE" -a -f "$IPPOOL_CONFIG_FILE" ]; then
318         write_loc_prop "ippool-config-file" $IPPOOL_CONFIG_FILE \
319             $CREATE_LOC_LEGACY_FILE
320     fi
321     if [ -n "$IKE_CONFIG_FILE" -a -f "$IKE_CONFIG_FILE" ]; then
322         write_loc_prop "ike-config-file" $IKE_CONFIG_FILE \
323             $CREATE_LOC_LEGACY_FILE
324     fi
325     if [ -n "$IPSEC_POLICY_CONFIG_FILE" -a -f "$IPSEC_POLICY_CONFIG_FILE" ]
326     then
327         write_loc_prop "ipsecpolicy-config-file" \
328             $IPSEC_POLICY_CONFIG_FILE $CREATE_LOC_LEGACY_FILE
329     fi
330
331     # End
332     echo "end" >> $CREATE_LOC_LEGACY_FILE
333     # network/location will create the Legacy location with these commands.
334 }
335
336 #
337 # Undoes the effects of the Legacy location creation
338 #
339 revert_to_legacy_loc () {
340     $SVCADM disable dns/client
341     $SVCADM disable nis/client
342     $SVCADM disable ldap/client
343
344     # copy nsswitch.conf to /etc/nsswitch.conf
345     copy_from_legacy_loc /etc/nsswitch.conf
346
347     # DNS - copy resolv.conf to /etc/resolv.conf
348     if [ -f "$PERM_LEGACY_PATH/resolv.conf" ]; then
349         copy_from_legacy_loc /etc/resolv.conf
350         $SVCADM enable dns/client
351     fi
352
353     # set /etc/defaultdomain and domainname(lm)
354     DEFAULT_DOMAIN='nwam_get_loc_prop Legacy default-domain'
355     if [ -n "$DEFAULT_DOMAIN" ]; then
356         $DOMAINNAME $DEFAULT_DOMAIN
357         $DOMAINNAME > /etc/defaultdomain
358     fi
359
360     # NIS - directory and ypserver in /var/yp/binding/
361     NIS_CONFIGSRC='nwam_get_loc_prop Legacy nis-nameservice-configsrc'
362     NIS_SERVERS='nwam_get_loc_prop Legacy nis-nameservice-servers'
363     if [ -n "$NIS_CONFIGSRC" ]; then
364         if [ ! -d "$NIS_BIND_PATH/$DEFAULT_DOMAIN" ]; then
365             $MKDIR -p $NIS_BIND_PATH/$DEFAULT_DOMAIN
366         fi

```

```

367         if [ -n "$NIS_SERVERS" ]; then
368             echo "$NIS_SERVERS" | $AWK \
369             echo "$NIS_SERVERS" | $NAWK \
370             'FS=" " { for (i = 1; i <= NF; i++) print $i }' \
371             > $NIS_BIND_PATH/$DEFAULT_DOMAIN/ypservers
372     fi
373     $SVCADM enable nis/client
374
375     # LDAP - copy ldap_client_file to /var/ldap/ldap_client_file
376     if [ -f "$PERM_LEGACY_PATH/ldap_client_file" ]; then
377         copy_from_legacy_loc /var/ldap/ldap_client_file
378         $SVCADM enable ldap/client
379     fi
380
381     # Copy back nfs NFSMAPID_DOMAIN
382     NFSMAPID_DOMAIN='nwam_get_loc_prop Legacy nfsv4-domain'
383     if [ -n "$NFSMAPID_DOMAIN" ]; then
384         set_smf_prop $NFS_MAPID_FMRI \
385             nfs-props/nfsmapid_domain $NFSMAPID_DOMAIN
386         $SVCADM refresh $NFS_MAPID_FMRI
387         $SVCADM enable $NFS_MAPID_FMRI
388     fi
389
390     # IPFilter, IPsec, and IKE
391     ipf_file='nwam_get_loc_prop Legacy ipfilter-config-file'
392     ipf6_file='nwam_get_loc_prop Legacy ipfilter-v6-config-file'
393     ipnat_file='nwam_get_loc_prop Legacy ipnat-config-file'
394     ippool_file='nwam_get_loc_prop Legacy ippool-config-file'
395     ike_file='nwam_get_loc_prop Legacy ike-config-file'
396     pol_file='nwam_get_loc_prop Legacy ipsecpolicy-config-file'
397
398     if [ -n "$ike_file" ]; then
399         copy_from_legacy_loc $ike_file
400         set_smf_prop $IPSEC_IKE_FMRI config/config_file $ike_file
401         $SVCADM refresh $IPSEC_IKE_FMRI
402         $SVCADM enable $IPSEC_IKE_FMRI
403     else
404         set_smf_prop $IPSEC_IKE_FMRI config/config_file \
405             $IPSEC_IKE_DEFAULT_CONFIG_FILE
406         $SVCADM disable $IPSEC_IKE_FMRI
407     fi
408     if [ -n "$pol_file" ]; then
409         copy_from_legacy_loc $pol_file
410         set_smf_prop $IPSEC_POLICY_FMRI config/config_file $pol_file
411         $SVCADM refresh $IPSEC_POLICY_FMRI
412         $SVCADM enable $IPSEC_POLICY_FMRI
413     else
414         set_smf_prop $IPSEC_POLICY_FMRI config/config_file \
415             $IPSEC_POLICY_DEFAULT_CONFIG_FILE
416         $SVCADM disable $IPSEC_POLICY_FMRI
417     fi
418
419     refresh_ipf=false
420     if [ -n "$ipf_file" ]; then
421         # change /none, /allow, and /deny to firewall policy
422         if [ "$ipf_file" = "/none" -o "$ipf_file" = "/allow" \
423             -o "$ipf_file" = "/deny" ]; then
424             policy='echo "$ipf_file" | $AWK 'FS="/" { print $2 }''
425             policy='echo "$ipf_file" | $NAWK 'FS="/" { print $2 }''
426             set_smf_prop $IPFILTER_FMRI \
427                 firewall_config_default/policy $policy
428             # no need to clear custom_policy_file as it isn't "custo
429         else
430             copy_from_legacy_loc $ipf_file
431             set_smf_prop $IPFILTER_FMRI \

```

```
431             firewall_config_default/policy "custom"
432             set_smf_prop $IPFILTER_FMRI \
433             firewall_config_default/custom_policy_file $ipf_file
434         fi
435         refresh_ipf=true
436     fi
437     if [ -n "$ipf6_file" ]; then
438         copy_from_legacy_loc $ipf6_file
439         set_smf_prop $IPFILTER_FMRI config/ipf6_config_file $ipf6_file
440         refresh_ipf=true
441     else
442         set_smf_prop $IPFILTER_FMRI config/ipf6_config_file \
443         $IPF6_DEFAULT_CONFIG_FILE
444     fi
445     if [ -n "$ipnat_file" ]; then
446         copy_from_legacy_loc $ipnat_file
447         set_smf_prop $IPFILTER_FMRI config/ipnat_config_file $ipnat_file
448         refresh_ipf=true
449     else
450         set_smf_prop $IPFILTER_FMRI config/ipnat_config_file \
451         $IPNAT_DEFAULT_CONFIG_FILE
452     fi
453     if [ -n "$ippool_file" ]; then
454         copy_from_legacy_loc $ippool_file
455         set_smf_prop $IPFILTER_FMRI config/ippool_config_file \
456         $ippool_file
457         refresh_ipf=true
458     else
459         set_smf_prop $IPFILTER_FMRI config/ippool_config_file \
460         $IPPOOL_DEFAULT_CONFIG_FILE
461     fi
462
463     $SVCADM refresh $IPFILTER_FMRI
464     if [ "$refresh_ipf" = "true" ]; then
465         $SVCADM enable $IPFILTER_FMRI
466     else
467         $SVCADM disable $IPFILTER_FMRI
468     fi
469
470     # Remove the Legacy directories, script and location
471     $RM -rf $VOL_LEGACY_PATH
472     $RM -rf $PERM_LEGACY_PATH
473     $RM -f $VOL_NWAM_PATH/create_loc_legacy
474     $NWAMCFG destroy loc Legacy
475 }
```

unchanged portion omitted

new/usr/src/cmd/svc/milestone/net-routing-setup

1

```
*****
7438 Sun May 26 18:22:50 2013
new/usr/src/cmd/svc/milestone/net-routing-setup
XXX Remove awk(1)
*****
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.

25 # This script configures IP routing.

27 . /lib/svc/share/smf_include.sh

29 #
30 # In a shared-IP zone we need this service to be up, but all of the work
31 # it tries to do is irrelevant (and will actually lead to the service
32 # failing if we try to do it), so just bail out.
33 # In the global zone and exclusive-IP zones we proceed.
34 #
35 smf_configure_ip || exit $SMF_EXIT_OK

37 #
38 # If routing.conf file is in place, and has not already been read in
39 # by previous invocation of routeadm, legacy configuration is upgraded
40 # by this call to "routeadm -u". This call is also needed when
41 # a /var/svc/profile/upgrade file is found, as it may contain routeadm commands
42 # which need to be applied. Finally, routeadm starts in.ndpd by
43 # enabling the ndp service (in.ndpd), which is required for IPv6 address
44 # autoconfiguration. It would be nice if we could do this in
45 # network/loopback, but since the SMF backend is read-only at that
46 # point in boot, we cannot.
47 #
48 /sbin/routeadm -u

50 #
51 # Are we routing dynamically? routeadm(1M) reports this in the
52 # "current" values of ipv4/6-routing - if either are true, we are running
53 # routing daemons (or at least they are enabled to run).
54 #
55 dynamic_routing_test='/sbin/routeadm -p | \
56 /usr/xpg4/bin/awk '/^ipv[46]-routing [.]*/ { print $2 }' | /usr/bin/grep "curre
57 awk '/^ipv[46]-routing [.]*/ { print $2 }' | /usr/bin/grep "current=enabled"
57 if [ -n "$dynamic_routing_test" ]; then
58     dynamic_routing="true"
59 fi
```

new/usr/src/cmd/svc/milestone/net-routing-setup

2

```
61 #
62 # Configure default IPv4 routers using the local "/etc/defaultrouter"
63 # configuration file. The file can contain the hostnames or IP
64 # addresses of one or more default routers. If hostnames are used,
65 # each hostname must also be listed in the local "/etc/hosts" file
66 # because NIS is not running at the time that this script is
67 # run. Each router name or address is listed on a single line by
68 # itself in the file. Anything else on that line after the router's
69 # name or address is ignored. Lines that begin with "#" are
70 # considered comments and ignored.
71 #
72 # The default routes listed in the "/etc/defaultrouter" file will
73 # replace those added by the kernel during diskless booting. An
74 # empty "/etc/defaultrouter" file will cause the default route
75 # added by the kernel to be deleted.
76 #
77 # Note that the default router file is ignored if we received routes
78 # from a DHCP server. Our policy is to always trust DHCP over local
79 # administration.
80 #
81 smf_netstrategy

83 if [ "$_INIT_NET_STRATEGY" = "dhcp" ] && \
84     [ -n "/sbin/dhcpinfo Router" ]; then
85     defrouters='/sbin/dhcpinfo Router'
86 elif [ -f /etc/defaultrouter ]; then
87     defrouters='/usr/bin/grep -v \#\# /etc/defaultrouter | \
88         /usr/bin/awk '{print $1}'
89     if [ -n "$defrouters" ]; then
90         #
91         # We want the default router(s) listed in
92         # /etc/defaultrouter to replace the one added from the
93         # BOOTPARAMS WHOAMI response but we must avoid flushing
94         # the last route between the running system and its
95         # /usr file system.
96         #
97         # First, remember the original route.
98         shift $#
99         set -- '/usr/bin/netstat -rn -f inet | \
100             /usr/bin/grep '^default''
101         route_IP="$2"
102         #
103         # Next, add those from /etc/defaultrouter. While doing
104         # this, if one of the routes we add is for the route
105         # previously added as a result of the BOOTPARAMS
106         # response, we will see a message of the form:
107         # "add net default: gateway a.b.c.d: entry exists"
108         #
109         do_delete=yes
110         for router in $defrouters; do
111             route_added='/usr/sbin/route -n add default \
112                 -gateway $router'
113             res=$?
114             set -- $route_added
115             [ $res -ne 0 -a "$5" = "$route_IP:" ] && do_delete=no
116         done
117         #
118         # Finally, delete the original default route unless it
119         # was also listed in the defaultrouter file.
120         #
121         if [ -n "$route_IP" -a $do_delete = yes ]; then
122             /usr/sbin/route -n delete default \
123                 -gateway $route_IP >/dev/null
```

new/usr/src/cmd/svc/milestone/net-routing-setup

3

```

127         fi
128     else
129         /usr/sbin/route -fn > /dev/null
130     fi
131 else
132     defrouters=
133 fi

135 #
136 # Use routeadm(1M) to configure forwarding and launch routing daemons
137 # for IPv4 and IPv6 based on preset values. These settings only apply
138 # to the global zone. For IPv4 dynamic routing, the system will default
139 # to disabled if a default route was previously added via BOOTP, DHCP,
140 # or the /etc/defaultrouter file. routeadm also starts in.ndpd.
141 #
142 if [ "$dynamic_routing" != "true" ] && [ -z "$defrouters" ]; then
143     #
144     # No default routes were setup by "route" command above.
145     # Check the kernel routing table for any other default
146     # routes.
147     #
148     /usr/bin/netstat -rn -f inet | \
149     /usr/bin/grep default >/dev/null 2>&1 && defrouters=yes
150 fi

152 #
153 # The routeadm/ipv4-routing-set property is true if the administrator
154 # has run "routeadm -e/-d ipv4-routing". If not, we revert to the
155 # appropriate defaults. We no longer run "routeadm -u" on every boot
156 # however, as persistent daemon state is now controlled by SMF.
157 #
158 ipv4_routing_set='/usr/bin/svcprop -p routeadm/ipv4-routing-set $SMF_FMRI`
159 if [ -z "$defrouters" ]; then
160     #
161     # Set default value for ipv4-routing to enabled. If routeadm -e/-d
162     # has not yet been run by the administrator, we apply this default.
163     # The -b option is project-private and informs routeadm not
164     # to treat the enable as administrator-driven.
165     #
166     /usr/sbin/svccfg -s $SMF_FMRI \
167     setprop routeadm/default-ipv4-routing = true
168     if [ "$ipv4_routing_set" = "false" ]; then
169         /sbin/routeadm -b -e ipv4-routing -u
170     fi
171 else
172     #
173     # Default router(s) have been found, so ipv4-routing default value
174     # should be disabled. If routeadm -e/d has not yet been run by
175     # the administrator, we apply this default. The -b option is
176     # project-private and informs routeadm not to treat the disable as
177     # administrator-driven.
178     #
179     /usr/sbin/svccfg -s $SMF_FMRI \
180     setprop routeadm/default-ipv4-routing = false
181     if [ "$ipv4_routing_set" = "false" ]; then
182         /sbin/routeadm -b -d ipv4-routing -u
183     fi
184 fi

186 #
187 # See if static routes were created by install. If so, they were created
188 # under /etc/svc/volatile. Copy them into their proper place.
189 #
190 if [ -f /etc/svc/volatile/etc/inet/static_routes ]; then
191     echo "Installing persistent routes"
192     if [ -f /etc/inet/static_routes ]; then

```

new/usr/src/cmd/svc/milestone/net-routing-setup

4

```

193         cat /etc/svc/volatile/etc/inet/static_routes | grep -v '^#' \
194         >> /etc/inet/static_routes
195     else
196         cp /etc/svc/volatile/etc/inet/static_routes \
197         /etc/inet/static_routes
198     fi
199     /usr/bin/rm /etc/svc/volatile/etc/inet/static_routes
200
201 fi

203 #
204 # Read /etc/inet/static_routes and add each route.
205 #
206 if [ -f /etc/inet/static_routes ]; then
207     echo "Adding persistent routes:"
208     /usr/bin/egrep -v "^(#|$)" /etc/inet/static_routes | while read line; do
209         /usr/sbin/route add $line
210     done
211 fi

213 # Clear exit status.
214 exit $SMF_EXIT_OK

```


new/usr/src/cmd/svc/milestone/net-svc

1

6897 Sun May 26 18:22:50 2013

new/usr/src/cmd/svc/milestone/net-svc

XXX Remove awk(1)

unchanged portion omitted

```
119 #
120 # update_resolv()
121 # Go through /etc/resolv.conf and replace any existing domain or
122 # nameserver entries with new ones derived from DHCP. Note that
123 # it is important to preserve order of domain entries vs. search
124 # entries; the search entries are reserved for administrator
125 # customization and if placed after the domain entry will override
126 # it. See resolv.conf(4).
127 #
128 # The first arg should be the dns servers string, the second
129 # should be the dns domain.
130 #
131 update_resolv ()
132 {
133     dnsservers=$1
134     dnsdomain=$2
135
136     if [ ! -f /etc/resolv.conf ]; then
137         /usr/bin/touch /etc/resolv.conf
138     fi
139     export dnsservers dnsdomain
140     /usr/xpg4/bin/awk </etc/resolv.conf >/tmp/resolv.conf.$$ '
140     /usr/bin/nawk </etc/resolv.conf >/tmp/resolv.conf.$$ '
141     function writedomain() {
142         if (updated == 0) {
143             # Use only first domain, not a search list
144             split(ENVIRON["dnsdomain"], d)
145             if(length(d[1]) != 0)
146                 printf("domain %s\n", d[1])
147         }
148         ++updated
149     }
150     $1 == "domain" { writedomain(); next }
151     $1 != "nameserver" { print $0 }
152     END {
153         writedomain()
154         n = split(ENVIRON["dnsservers"], s)
155         for (i = 1; i <= n; ++i)
156             printf("nameserver %s\n", s[i])
157     },
158     unset dnsservers dnsdomain
159     mv_file /tmp/resolv.conf.$$ /etc/resolv.conf 644
160 }
```

unchanged portion omitted

```
217 #
218 # Remove any lines with the "# Added by DHCP" tag from /etc/inet/hosts.
219 #
220 cleanup_hosts ()
221 {
222     /usr/xpg4/bin/awk '{
222     /usr/bin/nawk '{
223         if (index($0, "# Added by DHCP") == 0 ||
224             $1 == "127.0.0.1" || $1 == "::1") {
225             print $0
226         }
227     }' /etc/inet/hosts > /tmp/hosts.$$
228     mv_file /tmp/hosts.$$ /etc/inet/hosts 444
229 }
```

unchanged portion omitted

new/usr/src/cmd/svc/shell/net_include.sh

1

22362 Sun May 26 18:22:50 2013

new/usr/src/cmd/svc/shell/net_include.sh

XXX Remove awk(1)

unchanged_portion_omitted_

```
707 #
708 # Check for use of the default "Port VLAN Identifier" (PVID) -- VLAN 1.
709 # If there is one for a given interface, then warn the user and force the
710 # PVID to zero (if it's not already set). We do this by generating a list
711 # of interfaces with VLAN 1 in use first, and then parsing out the
712 # corresponding base datalink entries to check for ones without a
713 # "default_tag" property.
714 #
715 update_pvid()
716 {
717     datalink=/etc/dladm/datalink.conf
718
719     (
720         # Find datalinks using VLAN 1 explicitly
721         # configured by dladm
722         /usr/xpg4/bin/awk '
722         /usr/bin/nawk '
723         /^#/ || NF < 2 { next }
724         { linkdata[$1]=$2; }
725         /;vid=int,1;/ {
726             sub(/.*;linkover=int,/, "", $2);
727             sub(/.*/, "", $2);
728             link=linkdata[$2];
729             sub(/name=string,/, "", link);
730             sub(/.*/, "", link);
731             print link;
732         }' $datalink
733     ) | ( /usr/bin/sort -u; echo END; cat $datalink ) | /usr/xpg4/bin/awk '
733     ) | ( /usr/bin/sort -u; echo END; cat $datalink ) | /usr/bin/nawk '
734     /^END$/ { state=1; }
735     state == 0 { usingpvid[+usingpvid]=$1; next; }
736     /^#/ || NF < 2 { next; }
737     {
738         # If it is already present and has a tag set,
739         # then believe it.
740         if (!match($2, /;default_tag=/))
741             next;
742         sub(/name=string,/, "", $2);
743         sub(/.*/, "", $2);
744         for (i = 1; i <= usingpvid; i++) {
745             if (usingpvid[i] == $2)
746                 usingpvid[i]="";
747         }
748     }
749     END {
750         for (i = 1; i <= usingpvid; i++) {
751             if (usingpvid[i] != "") {
752                 printf("Warning: default VLAN tag set to 0" \
753                     " on %s\n", usingpvid[i]);
754                 cmd=sprintf("dladm set-linkprop -p " \
755                     "default_tag=0 %s\n", usingpvid[i]);
756                 system(cmd);
757             }
758         }
759     }'
760 }
```

unchanged_portion_omitted_

new/usr/src/cmd/svc/shell/routing_include.sh

1

7392 Sun May 26 18:22:51 2013

new/usr/src/cmd/svc/shell/routing_include.sh

XXX Remove nawk(1)

_____ unchanged_portion_omitted _____

```
152 #
153 # set_daemon_noption_properties inst_fmri optstring options propnames
154 #   default
155 #
156 # Function looks past option list for addition values, and sets properties
157 # specified in propnames to additional positional values. If no value
158 # is found for additional property, default is used.
159 #
160 set_daemon_noption_properties()
161 {
162     OPTIND=1
163     # Skip options
164     while getopts $3 opt $2; do
165         case $opt in
166             ? )
167                 esac
168             done
169             pos=$OPTIND
170             for prop in $4
171             do
172                 val='/usr/bin/echo $2 | /usr/xpg4/bin/awk -v POS=$pos \
172                 val='/usr/bin/echo $2 | /usr/bin/nawk -v POS=$pos \
173                 '{ print $POS }' \
174                 if [ -z "$val" ]; then
175                     val="$5"
176                 fi
177                 set_routeadm_property $1 $prop $val
178                 pos='expr $pos + 1'
179             done
180 }
```

```
182 #
183 # get_daemon_args $inst_fmri
184 #
185 # Retrieves routeadm/daemon-args property values, if any. Removes
186 # quotes around values including spaces.
```

```
187 #
188 get_daemon_args()
189 {
190     args='/usr/sbin/svccfg -s $1 listprop routeadm/daemon-args | \
191     /usr/xpg4/bin/awk '{ for (i = 3; i <= NF; i++) printf "%s ", $i }' | \
192     /usr/xpg4/bin/awk '{ sub(/\^"/, ""); sub(/\\"[ \t]*$/, ""); print }' \
193     /usr/bin/nawk '{ for (i = 3; i <= NF; i++) printf "%s ", $i }' | \
194     /usr/bin/nawk '{ sub(/\^"/, ""); sub(/\\"[ \t]*$/, ""); print }' \
195     echo "$args"
196 }
```

_____ unchanged_portion_omitted _____

```
206 #
207 # The functions below are used to map back from property settings to
208 # commandline arguments to launch daemons.
```

```
211 get_routeadm_property()
212 {
213     propval='/sbin/routeadm -l $1 | /usr/xpg4/bin/awk -v PROP=$2 \
213     propval='/sbin/routeadm -l $1 | /usr/bin/nawk -v PROP=$2 \
214     '($1 == PROP) { for (i = 3; i < NF; i++) printf $i " "; \
215     if (NF >= 3) {printf $NF}}'
```

new/usr/src/cmd/svc/shell/routing_include.sh

2

```
216     echo "$propval"
217 }
```

_____ unchanged_portion_omitted _____

```
247 #
248 # get_daemon_ordered_multivalue_option_from_property inst_fmri prop
249 # option
250 #
251 # Returns appropriate daemon option and associated values. Values are
252 # unquoted, i.e. -A value1 -A value2
253 #
254 get_daemon_ordered_multivalue_option_from_property()
255 {
256     # get property values, removing trailing delimiter.
257     propvals='get_routeadm_property $1 $2 | \
258     /usr/xpg4/bin/awk '{sub(/;[ \t]*$/, ""); print }' \
258     /usr/bin/nawk '{sub(/;[ \t]*$/, ""); print }' \
259     # Substitute switch for internal delimiters.
260     fixed_propvals='/usr/bin/echo $propvals | \
261     /usr/xpg4/bin/awk -v SWITCH=" -${3} " \
261     /usr/bin/nawk -v SWITCH=" -${3} " \
262     '{sub(/;/, SWITCH); print }' \
263     if [ -n "$fixed_propvals" ]; then
264         echo "-${3} $fixed_propvals"
265     fi
266 }
```

_____ unchanged_portion_omitted _____

```
new/usr/src/cmd/svr4pkg/pkgscripts/i.CompCpio.sh
```

1

```
*****
12735 Sun May 26 18:22:51 2013
new/usr/src/cmd/svr4pkg/pkgscripts/i.CompCpio.sh
XXX Remove awk(1)
*****
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 #
24 # Copyright 2005 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 #
28 # i.CompCpio
29 #
30 # This shell script uncompresses and installs files archived in
31 # old-style WOS packages using the utilities cpio and compress. It
32 # looks in the PKGSRC directory for the archives which may be called
33 # out in one of eight ways :
34 #
35 #     reloc.cpio.Z   relocatable paths, less old style
36 #     root.cpio.Z    absolute paths, less old style
37 #     reloc.cpio     relocatable paths less old style, not compressed
38 #     root.cpio      absolute paths, less old style, not compressed
39 #     reloc.Z        relocatable paths, old style, compressed
40 #     root.Z         absolute paths, old style, compressed
41 #     reloc          relocatable paths, old style, not compressed
42 #     root           absolute paths, old style, not compressed
43 #
44 # stdin carries the source directory as the first entry followed by the
45 # paths of the files to be installed as indicated in the pkgmap. Since
46 # all operations take place from the declared base directory, both relative
47 # and absolute paths will install correctly. There are three methods and
48 # since speed is of the essence, we skip straight to the right one :
49 #
50 #     If it's an initial install
51 #         do a full cpio for each archive
52 #     else
53 #         If there's only the reloc archive
54 #             make a file list, rm executables, do a selective cpio
55 #         else
56 #             rm executables, do a full cpio for each archive
57 #
58 # Since the old-style archives contain no execute permissions, this
59 # script saves the executables it requires so it can clean up after
60 # unloading the archive. If /usr/lib/ld.so or .so.1 is included in the
61 # package, no cleanup will be possible (nothing will run) so we clean
```

```
new/usr/src/cmd/svr4pkg/pkgscripts/i.CompCpio.sh
```

2

```
62 # up first and then unload the entire archive without a file list.
63 #
64 NAME="i.CompCpio"
65 FILELIST=${PKGSAV:?undefined}/filelist
66 BD=${BASEDIR:-/}
67 IR=${PKG_INSTALL_ROOT:-/}
68 MAXLIST=550 # This is arbitrary based upon 2.4 cpio
69 count=0
70 #
71 reloc_cpio_Z=0
72 root_cpio_Z=0
73 reloc_cpio=0
74 root_cpio=0
75 Reloc_Arch=""
76 Root_Arch=""
77 is_an_archive=0
78 is_a_filelist=0
79 mk_filelist=0
80 list_empty=1
81 local_install=0
82 Spcl_init=0
83 Rm_alt_sav=0
84 #
85 # critical archived dynamic libraries and executables
86 Spcl_lib=0
87 Spcl_exec=0
88 Movelist=""
89 Ld_Preload=""
90 Ldl=usr/lib/ld.so.1
91 Ld=usr/lib/ld.so
92 Libintl=usr/lib/libintl.so.1
93 Libmalloc=usr/lib/libmapmalloc.so.1
94 Libc=usr/lib/libc.so.1
95 Libw=usr/lib/libw.so.1
96 Libdl=usr/lib/libdl.so.1
97 Cpio=usr/bin/cpio
98 Rm=usr/bin/rm
99 Ln=usr/bin/ln
100 Mv=usr/bin/mv
101 Awk=usr/xpg4/bin/awk
101 Nawk=usr/bin/nawk
102 Zcat=usr/bin/zcat
103 #
104 # Set up the default paths
105 MV_xpath=/usr/bin
106 MV_cmd=$MV_xpath/mv
107 CPIO_xpath=/usr/bin
108 CPIO_cmd=$CPIO_xpath/cpio
109 ZCAT_xpath=/usr/bin
110 ZCAT_cmd=$ZCAT_xpath/zcat
111 LN_xpath=/usr/bin
112 LN_cmd=$LN_xpath/ln
113 AWK_xpath=/usr/xpg4/bin/
114 AWK_cmd=$AWK_xpath/awk
113 Nawk_xpath=/usr/bin
114 Nawk_cmd=$Nawk_xpath/nawk
115 RM_xpath=/usr/bin
116 RM_cmd=$RM_xpath/rm
117 Tmp_xpath=/usr/tmp$$dir
118 Tmp_Creat=0
119 rm_cpio=0
120 rm_ln=0
121 rm_zcat=0
122 rm_awk=0
122 rm_nawk=0
123 rm_rm=0
```

```

124 rm_mv=0
125 no_select=0

127 # Functions

129 #
130 # This creates the temporary directory for holding the old dynamic
131 # libraries and executables.
132 #
133 mktempdir() {
134     if [ ! -d $Tmp_xpath ]; then
135         mkdir $Tmp_xpath
136         if [ $? -ne 0 ]; then
137             echo `gettext "ERROR : $NAME cannot create $Tmp_xpath."`
138             exit 1
139         fi
140     fi
141     Tmp_Creat=1
142 }

144 #
145 # Test a path to see if it represents a dynamic library or executable that
146 # we use in this script. If it is, deal with the special case.
147 #
148 spclcase() { # $1 is the pathname to special case
149     if [ $local_install -eq 1 ]; then
150         case $1 in
151             $Ld)          no_select=1;;
152             $Ldl)        no_select=1;;
153             $Libintl)    Spcl_lib=1; file=libintl.so.1;;
154             $Libmalloc) Spcl_lib=1; file=libmmapalloc.so.1;;
155             $Libc)       Spcl_lib=1; file=libc.so.1;;
156             $Libw)       Spcl_lib=1; file=libw.so.1;;
157             $Libdl)      Spcl_lib=1; file=libdl.so.1;;
158             $Cpio)       rm_cpio=1; Spcl_exec=1;;
159             $Ln)         rm_ln=1; Spcl_exec=1;;
160             $Zcat)       rm_zcat=1; Spcl_exec=1;;
161             $Awk)        rm_awk=1; Spcl_exec=1;;
162             $Nawk)       rm_nawk=1; Spcl_exec=1;;
163             $Rm)         rm_rm=1; Spcl_exec=1;;
164             $Mv)         rm_mv=1; Spcl_exec=1;;
165         esac

166         if [ $no_select -eq 1 ]; then
167             is_a_filelist=0
168             list_empty=1
169             $RM_cmd $FILELIST
170             if [ $Rm_alt_sav -eq 1 ]; then
171                 $RM_cmd -r $PKGSAV
172                 Rm_alt_sav=0
173             fi
174             exec_clean 1
175             return 1
176         elif [ $Spcl_lib -eq 1 ]; then
177             if [ $Tmp_Creat -eq 0 ]; then
178                 mktempdir
179             fi

181             if [ $Spcl_init -eq 0 ]; then
182                 Org_LD_LIBRARY_PATH=${LD_LIBRARY_PATH}
183                 LD_LIBRARY_PATH="$Org_LD_LIBRARY_PATH $Tmp_xpath"
184                 export LD_LIBRARY_PATH
185                 Spcl_init=1
186             fi
187             Ld_Preload="$Ld_Preload $Tmp_xpath/$file"
188             LD_PRELOAD=$Ld_Preload

```

```

189     export LD_PRELOAD
190     Movelist="$1 $file $Movelist"
191     $MV_cmd $1 $Tmp_xpath
192     $LN_cmd -s ../../$Tmp_xpath/$file $1
193     Spcl_lib=0
194     elif [ $Spcl_exec -eq 1 ]; then
195         if [ $Tmp_Creat -eq 0 ]; then
196             mktempdir
197         fi

199     $MV_cmd $1 $Tmp_xpath
200     if [ $rm_cpio -eq 1 ]; then
201         $LN_cmd -s ../../$Tmp_xpath/cpio $1
202         CPIO_cmd="$Tmp_xpath/cpio"
203         Movelist="$1 cpio $Movelist"
204         rm_cpio=0
205     elif [ $rm_ln -eq 1 ]; then
206         $Tmp_xpath/ln -s ../../$Tmp_xpath/ln $1
207         LN_cmd="$Tmp_xpath/ln"
208         Movelist="$1 ln $Movelist"
209         rm_ln=0
210     elif [ $rm_awk -eq 1 ]; then
211         $LN_cmd -s ../../$Tmp_xpath/awk $1
212         AWK_cmd="$Tmp_xpath/awk"
213         Movelist="$1 awk $Movelist"
214         rm_awk=0
215     elif [ $rm_nawk -eq 1 ]; then
216         $LN_cmd -s ../../$Tmp_xpath/nawk $1
217         NAWK_cmd="$Tmp_xpath/nawk"
218         Movelist="$1 nawk $Movelist"
219         rm_nawk=0
220     elif [ $rm_zcat -eq 1 ]; then
221         $LN_cmd -s ../../$Tmp_xpath/zcat $1
222         ZCAT_cmd="$Tmp_xpath/zcat"
223         Movelist="$1 zcat $Movelist"
224         rm_zcat=0
225     elif [ $rm_rm -eq 1 ]; then
226         $LN_cmd -s ../../$Tmp_xpath/rm $1
227         RM_cmd="$Tmp_xpath/rm"
228         Movelist="$Movelist $1 rm"
229         rm_rm=0
230     elif [ $rm_mv -eq 1 ]; then
231         $LN_cmd -s ../../$Tmp_xpath/mv $1
232         MV_cmd="$Tmp_xpath/mv"
233         Movelist="$Movelist $1 mv"
234         rm_mv=0
235     fi
236     Spcl_exec=0
237 fi
238 fi

239 return 0
240 }

241 #
242 # Clean up the libraries and executables that were moved.
243 #
244 exec_clean() { # $1 = 1 means be quiet
245     if [ ! -z "${Movelist}" ]; then
246         echo $Movelist | $AWK_cmd '
247         echo $Movelist | $NAWK_cmd '
248         { split ($0, line)
249           for (n=1; n <= NF; n++) {
250             print line[n]
251           }
252         }' | while read path; do

```

```

249         read file
250         if [ -h $path ]; then # If it's our slink
251             # then put the original back
252             if [ $! -eq 0 ]; then
253                 echo 'gettext "WARNING : $path not found'
254             fi
255             $MV_cmd $Tmp_xpath/$file $path
256         else # if the archive put something down
257             # remove the temporary copy
258             $RM_cmd $Tmp_xpath/$file
259         fi
260     done
261     for path in $Movelist; do
262         if [ -x $path ]; then
263             case $path in
264                 $Cpio) CPIO_cmd="$CPIO_xpath/cpio";;
265                 $Ln) LN_cmd="$LN_xpath/ln";;
266                 $Zcat) ZCAT_cmd="$ZCAT_xpath/zcat";;
267                 $Awk) AWK_cmd="$AWK_xpath/awk";;
268                 $Nawk) NAWK_cmd="$NAWK_xpath/nawk";;
269                 $Rm) RM_cmd="$RM_xpath/rm";;
270                 $Mv) MV_cmd="$MV_xpath/mv";;
271             esac
272         fi
273     done
274     Movelist=""
275
276     if [ $Tmp_Creat -eq 1 ]; then
277         $RM_cmd -r $Tmp_xpath
278         Tmp_Creat=0
279     fi
280 }

```

unchanged portion omitted

```

351 #
352 # main
353 #
354
355 eval_pkg
356
357 if [ $BD = "/" ]; then
358     Client_BD=""
359 else
360     Client_BD='echo $BD | sed s@/@@'
361 fi
362
363 if [ $is_an_archive -eq 0 ]; then
364     echo 'gettext "ERROR : $NAME cannot find archived files in $PKGSRC."'
365     exit 1
366 fi
367
368 if [ ! -d $PKGSAV ]; then
369     echo 'gettext "WARNING : $NAME cannot find save directory $PKGSAV."'
370     PKGSAV=/tmp/$PKG.sav
371
372     if [ ! -d $PKGSAV ]; then
373         /usr/bin/mkdir $PKGSAV
374     fi
375
376     if [ $? -eq 0 ]; then
377         echo 'gettext " Using alternate save directory" $PKGSAV'
378         FILELIST=$PKGSAV/filelist
379         Rm_alt_sav=1
380     else
381         echo 'gettext "ERROR : cannot create alternate save directory"'

```

```

382         exit 1
383     fi
384 fi
385
386 if [ -f $FILELIST ]; then
387     rm $FILELIST
388 fi
389
390 cd $BD
391
392 # If there's one old-style archive and it is relocatable and this is
393 # not an initial install then make a file list for extraction.
394 if [ $is_an_archive -eq 1 -a ${PKG_INIT_INSTALL:-null} = null ]; then
395     mk_filelist=1
396 fi
397
398 # If this is not an initial install then clear out potentially executing
399 # files and libraries for cpio and create an extraction list if necessary
400 if [ ${PKG_INIT_INSTALL:-null} = null ]; then
401     if [ $local_install -eq 1 ]; then
402         # If extraction list is desired, create it
403         if [ $mk_filelist -eq 1 ]; then
404             is_a_filelist=1
405             while read path
406             do
407                 echo $path >> $FILELIST
408                 list_empty=0
409                 if [ -x ${path:-NULL} ]; then
410                     full_path='echo $Client_BD/$path | sed s
411                     spclcase $full_path
412                     if [ $? -eq 1 ]; then
413                         break
414                     fi
415                 fi
416             done
417
418             # If there's a path containing a '$' then we can't
419             # use the extraction list because of the shell
420             if [ $list_empty -eq 0 ]; then
421                 s='LD_PRELOAD="$Ld_Preload" $AWK_cmd ' /\$/ { p
422                 s='LD_PRELOAD="$Ld_Preload" $NAWK_cmd ' /\$/ {
423
424                 if [ ! -z "${s}" ]; then
425                     is_a_filelist=0
426                 fi
427             else # No extraction list is desired
428                 while read path
429                 do
430                     if [ -x ${path:-NULL} ]; then
431                         full_path='echo $Client_BD/$path | sed s
432                         spclcase $full_path
433                         if [ $? -eq 1 ]; then
434                             break
435                         fi
436                     fi
437                 done
438                 fi # $mk_filelist -eq 1
439             else # ! ($local_install -eq 1)
440                 # If extraction list is desired, create it
441                 if [ $mk_filelist -eq 1 ]; then
442                     is_a_filelist=1
443                     while read path
444                     do
445                         echo $path >> $FILELIST
446                         list_empty=0

```

```

447         done
449         # If there's a path containing a '$' then we can't
450         # use the extraction list because of the shell
451         if [ $!list_empty -eq 0 ]; then
452             s='LD_PRELOAD="$Ld_Preload" $AWK_cmd ' /\$/ { p
453             s='LD_PRELOAD="$Ld_Preload" $NAWK_cmd ' /\$/ {
454
455         if [ ! -z "${s}" ]; then
456             is_a_filelist=0
457         fi
458         # $mk_filelist -eq 1
459         fi # $local_install -eq 1
460 fi # ${PKG_INIT_INSTALL:-null} = null
462 # Now extract the data from the archive(s)
463 # extract compressed cpio relocatable archive
464 if [ $reloc_cpio_Z -eq 1 ]; then
465     cd $BD
466     if [ $is_a_filelist -eq 1 ]; then
467         if [ $!list_empty -eq 0 ]; then
468             $ZCAT_cmd $Reloc_Arch | $CPIO_cmd -idukm -E $FILELIST
469             if [ $? -ne 0 ]; then
470                 echo 'gettext "cpio of $Reloc_Arch failed with e
471                 exit 1
472             fi
473         fi
474     else
475         $ZCAT_cmd $Reloc_Arch | $CPIO_cmd -idukm
476     fi
477 fi
478 fi
480 # extract compressed cpio absolute archive
481 if [ $root_cpio_Z -eq 1 ]; then
482     cd $IR
483     $ZCAT_cmd $Root_Arch | $CPIO_cmd -idukm
484     if [ $? -ne 0 ]; then
485         echo 'gettext "cpio of $Root_Arch failed with error $?."
486         exit 1
487     fi
488 fi
490 # extract cpio relocatable archive
491 if [ $reloc_cpio -eq 1 ]; then
492     cd $BD
493     if [ $is_a_filelist -eq 1 ]; then
494         if [ $!list_empty -eq 0 ]; then
495             $CPIO_cmd -idukm -I $Reloc_Arch -E $FILELIST
496
497             if [ $? -ne 0 ]; then
498                 echo 'gettext "cpio of $Reloc_Arch failed with e
499                 exit 1
500             fi
501         fi
502     else
503         $CPIO_cmd -idukm -I $Reloc_Arch
504     fi
505 fi
507 # extract cpio absolute archive
508 if [ $root_cpio -eq 1 ]; then
509     cd $IR
510     $CPIO_cmd -idukm -I $Root_Arch
511     if [ $? -ne 0 ]; then

```

```

512         echo 'gettext "cpio of $Root_Arch failed with error $?."
513         exit 1
514     fi
515 fi
517 if [ -f $FILELIST ]; then
518     $RM_cmd $FILELIST
519 fi
521 if [ $Rm_alt_sav -eq 1 ]; then
522     $RM_cmd -r $PKGSAV
523     Rm_alt_sav=0
524 fi
526 exec_clean 0
528 if [ $Tmp_Creat -eq 1 ]; then
529     $RM_cmd -r $Tmp_xpath
530 fi
532 if [ $Spcl_init -eq 1 ]; then
533     LD_LIBRARY_PATH=$Org_LD_LIBRARY_PATH
534     export LD_LIBRARY_PATH
535     Spcl_init=0
536 fi
538 exit 0

```

new/usr/src/cmd/svr4pkg/pkgscripts/i.awk.sh

1

```
*****
1726 Sun May 26 18:22:52 2013
new/usr/src/cmd/svr4pkg/pkgscripts/i.awk.sh
XXX Remove nawk(1)
*****
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 #
24 # Copyright 2002 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 #
28 error=no
29 while read src dest
30 do
31     [ "$src" = /dev/null ] && continue
32
33     echo "Modifying $dest"
34
35     # Strip PKG_INSTALL_ROOT from dest if installation is to an
36     # alternate root.
37
38     if [ -n "$PKG_INSTALL_ROOT" -a "$PKG_INSTALL_ROOT" != "/" ]; then
39         client_dest='echo $dest | \
40             /usr/xpg4/bin/awk -v rootdir="$PKG_INSTALL_ROOT" '{
41             /usr/bin/nawk -v rootdir="$PKG_INSTALL_ROOT" '{
42                 { print substr($0, length(rootdir)+1) }'`
43         savepath=$PKGSAB/awk${client_dest}
44     else
45         savepath=$PKGSAB/awk${dest}
46     fi
47
48     dirname='dirname $savepath'
49     if [ $? -ne 0 ]
50     then
51         error=yes
52         continue
53     fi
54     if [ ! -d $dirname ]
55     then
56         # ignore return code since mkdir has bug
57         mkdir -p $dirname
58     fi
59
60     cp $src $savepath &&
61     /usr/sadm/install/scripts/cmdexec /usr/bin/awk install $savepath $dest
```

new/usr/src/cmd/svr4pkg/pkgscripts/i.awk.sh

2

```
62     if [ $? -ne 0 ]
63     then
64         error=yes
65         continue
66     fi
67 done
68 [ "$error" = yes ] &&
69     exit 2
70 exit 0
```


new/usr/src/cmd/svr4pkg/pkgscripts/i.build.sh

1

```
*****
1836 Sun May 26 18:22:52 2013
new/usr/src/cmd/svr4pkg/pkgscripts/i.build.sh
XXX Remove awk(1)
*****
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 #
24 # Copyright 2002 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 #
28 error=no
29 while read src dest
30 do
31     [ "$src" = /dev/null ] && continue
32
33     echo "Modifying $dest"
34
35     # Strip PKG_INSTALL_ROOT from dest if installation is to an
36     # alternate root.
37
38     if [ -n "$PKG_INSTALL_ROOT" -a "$PKG_INSTALL_ROOT" != "/" ]; then
39         client_dest=`echo $dest | \
40             /usr/xpg4/bin/awk -v rootdir="$PKG_INSTALL_ROOT" '{
41             /usr/bin/nawk -v rootdir="$PKG_INSTALL_ROOT" '{
42                 { print substr($0, length(rootdir)+1) }'`
43         savepath=$PKGSAB/build${client_dest}
44     else
45         savepath=$PKGSAB/build${dest}
46     fi
47
48     dirname=`dirname $savepath`
49     if [ $? -ne 0 ]
50     then
51         error=yes
52         continue
53     fi
54
55     if [ ! -d $dirname ]
56     then
57         # ignore return since mkdir has bug
58         mkdir -p $dirname
59     fi
60
61     cp $src $savepath &&
```

new/usr/src/cmd/svr4pkg/pkgscripts/i.build.sh

2

```
61         chmod 500 $savepath
62     if [ $? -ne 0 ]
63     then
64         error=yes
65         continue
66     fi
67
68     if $savepath install > /tmp/$$build
69     then
70         if [ -s /tmp/$$build ]
71         then
72             cp /tmp/$$build $dest || error=yes
73         fi
74     else
75         error=yes
76     fi
77     rm -f /tmp/$$build
78 done
79 [ "$error" = yes ] &&
80     exit 2
81 exit 0
```

new/usr/src/cmd/svr4pkg/pkgscripts/i.sed.sh

1

1709 Sun May 26 18:22:53 2013

new/usr/src/cmd/svr4pkg/pkgscripts/i.sed.sh

XXX Remove nawk(1)

```
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 #
24 # Copyright 2002 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 #
28 error=no
29 while read src dest
30 do
31     [ "$src" = /dev/null ] && continue
32
33     echo "Modifying $dest"
34
35     # Strip PKG_INSTALL_ROOT from dest if installation is to an
36     # alternate root.
37
38     if [ -n "$PKG_INSTALL_ROOT" -a "$PKG_INSTALL_ROOT" != "/" ]; then
39         client_dest=`echo $dest | \
40             /usr/xpg4/bin/awk -v rootdir="$PKG_INSTALL_ROOT" '{
41             /usr/bin/nawk -v rootdir="$PKG_INSTALL_ROOT" '{
42                 { print substr($0, length(rootdir)+1) }'`
43         savepath=$PKGSAB/sed${client_dest}
44     else
45         savepath=$PKGSAB/sed${dest}
46     fi
47
48     dirname=`dirname $savepath`
49     if [ $? -ne 0 ]
50     then
51         error=yes
52         continue
53     fi
54
55     if [ ! -d $dirname ]
56     then
57         # ignore return since mkdir has bug
58         mkdir -p $dirname
59     fi
60
61     cp $src $savepath &&
```

new/usr/src/cmd/svr4pkg/pkgscripts/i.sed.sh

2

```
61         /usr/sadm/install/scripts/cmdexec /bin/sed install $savepath $de
62
63     if [ $? -ne 0 ]
64     then
65         error=yes
66     fi
67 done
68 [ "$error" = yes ] &&
69     exit 2
70 exit 0
```

new/usr/src/cmd/svr4pkg/pkgscripts/r.awk.sh

1

1625 Sun May 26 18:22:53 2013

new/usr/src/cmd/svr4pkg/pkgscripts/r.awk.sh

XXX Remove awk(1)

```
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 #
24 # Copyright 2002 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 #
28 error=no
29 while read dest
30 do
31     if [ -d $dest ]
32     then
33         echo "$dest"
34         rmdir $dest || error=yes
35     elif [ -f $dest ]
36     then
37         echo "Modifying $dest"
38
39         # Strip PKG_INSTALL_ROOT from dest if installation is to an
40         # alternate root.
41
42         if [ -n "$PKG_INSTALL_ROOT" -a "$PKG_INSTALL_ROOT" != "/" ]; then
43             client_dest=`echo $dest | \
44 /usr/xpg4/bin/awk -v rootdir="$PKG_INSTALL_ROOT"
44 /usr/bin/nawk -v rootdir="$PKG_INSTALL_ROOT" '{
45 { print substr($0, length(rootdir)+1) }'`
46             savepath=$PKGSAB/awk${client_dest}
47         else
48             savepath=$PKGSAB/awk${dest}
49         fi
50
51         /usr/sadm/install/scripts/cmdexec /usr/bin/awk remove $savepath
52         error=yes
53     else
54         [ -r $dest ] && echo "$dest"
55         rm -f $dest || error=yes
56     fi
57 done
58 [ "$error" = yes ] &&
59     exit 2
60 exit 0
```

new/usr/src/cmd/svr4pkg/pkgscripts/r.build.sh

1

```
*****
1757 Sun May 26 18:22:54 2013
new/usr/src/cmd/svr4pkg/pkgscripts/r.build.sh
XXX Remove awk(1)
*****
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 #
24 # Copyright 2002 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 #
28 error=no
29 while read dest
30 do
31     if [ -d $dest ]
32     then
33         echo "$dest"
34         rmdir $dest || error=yes
35     elif [ -f $dest ]
36     then
37         echo "Modifying $dest"
38     fi
39     # Strip PKG_INSTALL_ROOT from dest if installation is to an
40     # alternate root.
41     #
42     if [ -n "$PKG_INSTALL_ROOT" -a "$PKG_INSTALL_ROOT" != "/" ]; then
43         client_dest=`echo $dest | \
44             /usr/xpg4/bin/awk -v rootdir="$PKG_INSTALL_ROOT"
44             /usr/bin/nawk -v rootdir="$PKG_INSTALL_ROOT" '{
45                 { print substr($0, length(rootdir)+1)} }'`
46         savepath=$PKGSAB/build${client_dest}
47     else
48         savepath=$PKGSAB/build${dest}
49     fi
50     #
51     chmod +x $savepath
52     if $savepath remove > /tmp/$$build
53     then
54         if [ ! -s /tmp/$$build ]
55         then
56             rm -f $dest
57         else
58             cp /tmp/$$build $dest || error=yes
59         fi
60     else
```

new/usr/src/cmd/svr4pkg/pkgscripts/r.build.sh

2

```
61         error=yes
62     fi
63     rm -f /tmp/$$build
64     else
65         [ -r $dest ] && echo "$dest"
66         rm -f $dest || error=yes
67     fi
68 done
69 [ "$error" = yes ] &&
70     exit 2
71 exit 0
```

new/usr/src/cmd/svr4pkg/pkgscripts/r.sed.sh

1

1621 Sun May 26 18:22:54 2013

new/usr/src/cmd/svr4pkg/pkgscripts/r.sed.sh

XXX Remove awk(1)

```
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 #
24 # Copyright 2002 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 #
28 error=no
29 while read dest
30 do
31     if [ -d $dest ]
32     then
33         echo "$dest"
34         rmdir $dest || error=yes
35     elif [ -f $dest ]
36     then
37         echo "Modifying $dest"
38
39         # Strip PKG_INSTALL_ROOT from dest if installation is to an
40         # alternate root.
41
42         if [ -n "$PKG_INSTALL_ROOT" -a "$PKG_INSTALL_ROOT" != "/" ]; then
43             client_dest=`echo $dest | \
44                 /usr/xpg4/bin/awk -v rootdir="$PKG_INSTALL_ROOT"
44                 /usr/bin/nawk -v rootdir="$PKG_INSTALL_ROOT" '{
45                     { print substr($0, length(rootdir)+1) }'`
46             savepath=$PKGSAB/sed${client_dest}
47         else
48             savepath=$PKGSAB/sed${dest}
49         fi
50
51         /usr/sadm/install/scripts/cmdexec /bin/sed remove $savepath $des
52         error=yes
53     else
54         [ -r $dest ] && echo "$dest"
55         rm -f $dest || error=yes
56     fi
57 done
58 [ "$error" = yes ] &&
59     exit 2
60 exit 0
```

new/usr/src/cmd/tsol/demo/runwlabel.ksh

1

1495 Sun May 26 18:22:54 2013

new/usr/src/cmd/tsol/demo/runwlabel.ksh

XXX Remove awk(1)

```
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
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 #
27 # Usage:
28 # runwlabel "my-label" my-program
29 #
30 [ ! -x /usr/sbin/zoneadm ] && exit 0 # SUNWzoneu not installed
31
32 PATH=/usr/sbin:/usr/bin; export PATH
33
34 # Get the zone path associated with the "my-label" zone
35 # Remove the trailing "/root"
36 zonepath=`getzonepath "$1" | sed -e 's/\/root$//'`
37 progname="$2"
38
39 # Find the zone name that is associated with this zone path
40 for zone in `zoneadm list -pi | /usr/xpg4/bin/awk -F: -v zonepath=${zonepath} '{
41   if ("$4" == "${zonepath}") {
42     print $2
43   }
44 }'`; do
  unchanged_portion_omitted
```

new/usr/src/cmd/tsol/zones/zoneshare.sh

1

```
*****
2135 Sun May 26 18:22:55 2013
new/usr/src/cmd/tsol/zones/zoneshare.sh
XXX Remove nawk(1)
*****
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
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 #
25 # zoneshare -- share zone resources

27 # Processes a non-global zone's dfstab file after translating
28 # its zone-relative pathnames to global zone pathnames

30 USAGE="zoneshare -z zonename [- | file]"
31 set -- `getopt z: $*`
32 if [ $? != 0 ] # invalid options
33 then
34     echo $USAGE >&2
35     exit 1
36 fi
37 for i in $* # pick up the options
38 do
39     case $i in
40         -z) zonename=$2; shift 2;;
41         --) shift; break;;
42         esac
43 done

45 zoneattr='/usr/sbin/zoneadm -z $zonename list -p 2> /dev/null'
46 if [ $? -ne 0 ] # invalid zone
47 then
48     echo $USAGE >&2
49     exit 1
50 fi

52 prefix='echo $zoneattr | cut -d ":" -f4'
53 rootpath=$prefix/root

55 if [ $# -gt 1 ] # accept only one argument
56 then
57     echo $USAGE >&2
58     exit 1
59 elif [ $# = 1 ]
```

new/usr/src/cmd/tsol/zones/zoneshare.sh

2

```
60 then
61     case $1 in
62         -)     infile=;; # use stdin
63         *)     infile=$1;; # use a given source file
64         esac
65     else
66         infile=$prefix/etc/dfs/dfstab # default
67     fi

69 # Prepend each exported pathname with the zone's rootpath
70 # Skip over comments and lines without a share command
71 # Run each share command in its own shell

73 while read line # get complete lines
74 do
75     echo $line
76 done < $infile |
77 `egrep -v "^(#[#])" | /usr/xpg4/bin/awk -v rootpath=$rootpath \
79 `egrep -v "^(#[#])" | nawk -v rootpath=$rootpath \
78 '/share/ { ORS = " "; for (i = 1; i < NF; i++) print $i ; \
79     print rootpath $NF ";" } ' | /sbin/sh`
```

new/usr/src/cmd/tsol/zones/zoneunshare.sh

1

```
*****
1922 Sun May 26 18:22:55 2013
new/usr/src/cmd/tsol/zones/zoneunshare.sh
XXX Remove awk(1)
*****
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
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 #
27 # zoneunshare -- unshare zone resources
28 #
29 #
30 USAGE="zoneunshare -z zonename [- | file]"
31 set -- `getopt z: $*`
32 if [ $? != 0 ] # invalid options
33 then
34     echo $USAGE >&2
35     exit 1
36 fi
37 for i in $* # pick up the options
38 do
39     case $i in
40         -z) zonename=$2; shift 2;;
41         --) shift; break;;
42         esac
43 done
44 #
45 zoneattr='/usr/sbin/zoneadm -z $zonename list -p 2> /dev/null'
46 if [ $? -ne 0 ] # invalid zone
47 then
48     echo $USAGE >&2
49     exit 1
50 fi
51 #
52 prefix='echo $zoneattr | cut -d ":" -f4'
53 rootpath=$prefix/root
54 #
55 if [ $# -gt 1 ] # accept only one argument
56 then
57     echo $USAGE >&2
58     exit 1
59 elif [ $# = 1 ]
```

new/usr/src/cmd/tsol/zones/zoneunshare.sh

2

```
60 then
61     case $1 in
62         -) infile=;; # use stdin
63         *) infile=$1;; # use a given source file
64         esac
65     else
66         infile=/etc/dfs/sharetab # default
67     fi
68 #
69 # Run unshare for each resource in its own shell
70 #
71 while read line # get complete lines
72 do
73     echo $line
74 done < $infile |
75     'egrep "^$rootpath" | /usr/xpg4/bin/awk '{ print "/usr/sbin/unshare " $1 "'
77     'egrep "$rootpath" | /usr/sbin/awk '{ print "/usr/sbin/unshare " $1 ";" }' | /sbin/s
```


new/usr/src/cmd/ypcmd/multi.awk.sh

1

```
*****
3393 Sun May 26 18:22:56 2013
new/usr/src/cmd/ypcmd/multi.awk.sh
XXX Remove awk(1)
*****
1 #!/usr/xpg4/bin/awk -f
1 #!/usr/bin/nawk -f
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License, Version 1.0 only
7 # (the "License"). You may not use this file except in compliance
8 # with the License.
9 #
10 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
11 # or http://www.opensolaris.org/os/licensing.
12 # See the License for the specific language governing permissions
13 # and limitations under the License.
14 #
15 # When distributing Covered Code, include this CDDL HEADER in each
16 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
17 # If applicable, add the following below this CDDL HEADER, with the
18 # fields enclosed by brackets "[]" replaced with your own identifying
19 # information: Portions Copyright [yyyy] [name of copyright owner]
20 #
21 # CDDL HEADER END
22 #
23 #
24 #ident "%Z%M% %I% %E% SMI"
25 #
26 # Copyright (c) 1996, by Sun Microsystems, Inc.
27 # All rights reserved.
28 #
29 #
30 #
31 {
32 # Here we loop through the list of hostnames
33 # doing two separate things...
34 # First, we're building a list of hostnames
35 # for the current IP address ($1).
36 # Second, if we've seen a name before then
37 # we add the current address ($1) to a list
38 # of address associated with this particular
39 # name ($i).
40 #
41 # Note, that we're pretty careful about keeping
42 # out duplicates (and this has a cost).
43 #
44 for (i = 2; i <= NF; i++) {
45 # Make the namelist for this address
46 if (namelist[$1] == "") {
47 namelist[$1] = $i;
48 } else if (namelist[$1] == $i) {
49 ;
50 } else if (index(namelist[$1], $i) == 0) {
51 namelist[$1] = namelist[$1] " " $i;
52 } else {
53 nf = 1;
54 numnames = split(namelist[$1], n);
55 for (j = 1; j <= numnames; j++) {
56 if (n[j] == $i) {
57 nf = 0;
58 break;

```

new/usr/src/cmd/ypcmd/multi.awk.sh

2

```
59 }
60 }
61 if (nf) {
62 namelist[$1] = namelist[$1] " " $i;
63 nf = 0;
64 }
65 }
66 # Do we have an address for this name?
67 # If not, and it's not already there, add it.
68 if (addr[$i] == "") {
69 addr[$i] = $1;
70 } else if (index(addr[$i], $1) == 0) {
71 addr[$i] = addr[$i] "," $1
72 }
73 }
74 }
75 }
```

unchanged portion omitted

new/usr/src/cmd/ypcmd/net_files/Makefile

1

```
*****
19093 Sun May 26 18:22:56 2013
new/usr/src/cmd/ypcmd/net_files/Makefile
XXX Remove awk(1)
*****
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) 1990, 2010, Oracle and/or its affiliates. All rights reserved.
22 #
23 # Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989 AT&T
24 # All Rights Reserved
25 #
26 # Portions of this source code were derived from Berkeley
27 # under license from the Regents of the University of
28 # California.
29 #
30 #---
31 # It is somewhat confusing to note that Solaris 2.x uses /etc/auto_master
32 # instead of the 4.x /etc/auto.master file name because of NIS+ treating a
33 # "." in a special way.
34 #
35 # Set the following variable to "-b" to have NIS servers use the domain name
36 # resolver for hosts not in the current domain.
37 #B=-b
38 B=
39 DIR =/etc
40 #
41 # If the ipnodes (IPv6 hosts file) lives in a directory other than
42 # /etc/inet, then you'll need to change the following line.
43 #
44 INETDIR=/etc/inet
45 #
46 # If the auth_attr, exec_attr, prof_attr files
47 # live in a directory other than /etc/security, then you'll
48 # need to change the following line.
49 #
50 RBACDIR=/etc/security
51 #
52 # If the passwd, shadow and/or adjunct files used by rpc.yppasswdd
53 # live in directory other than /etc then you'll need to change the
54 # following line.
55 # DO NOT indent the line, however, since /etc/init.d/yp attempts
56 # to find it with grep "^PWDIR" ...
57 #
58 PWDIR =/etc
59 DOM = `domainname`
60 NOPUSH = ""
61 ALIASES = /etc/mail/aliases
```

new/usr/src/cmd/ypcmd/net_files/Makefile

2

```
62 YPDIR=/usr/lib/netsvc/yp
63 SBINDIR=/usr/sbin
64 YPDBDIR=/var/yp
65 YPPUSH=$(YPDIR)/yppush
66 MAKEDBM=$(SBINDIR)/makedbm
67 MULTI=$(YPDIR)/multi
68 REVNETGROUP=$(SBINDIR)/revnetgroup
69 STDETHERS=$(YPDIR)/stdethers
70 STDHOSTS=$(YPDIR)/stdhosts
71 MKNETID=$(SBINDIR)/mknetid
72 MKALIAS=$(YPDIR)/mkalias

74 CHKPIPE= || ( echo "NIS make terminated:" $@ 1>&2; kill -TERM 0 )

77 k:
78     @if [ ! $(NOPUSH) ]; then $(MAKE) $(MFLAGS) -k all; \
79     else $(MAKE) $(MFLAGS) -k all NOPUSH=$(NOPUSH);fi

81 all: passwd group hosts ipnodes ethers networks rpc services protocols \
82     netgroup bootparams aliases publickey netid netmasks c2secure \
83     timezone auto.master auto.home ageing \
84     auth.attr exec.attr prof.attr user.attr

86 c2secure:
87     -@if [ -f $(PWDIR)/security/passwd.adjunct ]; then \
88     if [ ! $(NOPUSH) ]; then $(MAKE) $(MFLAGS) -k \
89     passwd.adjunct.time group.adjunct.time; \
90     else $(MAKE) $(MFLAGS) -k NOPUSH=$(NOPUSH) \
91     passwd.adjunct.time group.adjunct.time; \
92     fi; \
93     fi

95 passwd.time: $(PWDIR)/passwd $(PWDIR)/shadow
96     -@if [ -f $(PWDIR)/security/passwd.adjunct ]; then \
97     (/usr/xpg4/bin/awk 'BEGIN { FS=":"; OFS=":" } /^[a-zA-Z0-9_]/ {
98     (/usr/xpg4/bin/awk 'BEGIN { FS=":"; OFS=":" } /^[a-zA-Z0-9_]/ {
99     (awk 'BEGIN { FS=":"; OFS=":" } /^[a-zA-Z0-9_]/ { $$2 = "##" $$
100     (awk 'BEGIN { FS=":"; OFS=":" } /^[a-zA-Z0-9_]/ { $$2 = "##" $$
101     (awk 'BEGIN { FS=":"; OFS=":" } /^[a-zA-Z0-9_]/ { $$2 = "##" $$
102     else \
103     (awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { print $$1, $
104     (awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { printf("%-10
105     fi
106     @touch passwd.time;
107     @echo "updated passwd";
108     @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) passwd.byname; fi
109     @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) passwd.byuid; fi
110     @if [ ! $(NOPUSH) ]; then echo "pushed passwd"; fi

112 group.time: $(DIR)/group
113     @(awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { print $$1, $$0 }' $
114     @(awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { printf("%-10d ", $$
115     @touch group.time;
116     @echo "updated group";
117     @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) group.byname; fi
118     @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) group.bygid; fi
119     @if [ ! $(NOPUSH) ]; then echo "pushed group"; fi

121 project.time: $(DIR)/project
122     @(awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { print $$1, $$0 }' $
123     @(awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { printf("%-10d ", $$
```

```

124 @touch project.time;
125 @echo "updated project";
126 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) project.byname; fi
127 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) project.byprojid; fi
128 @if [ ! $(NOPUSH) ]; then echo "pushed project"; fi

130 ipnodes.time: $(INETDIR)/ipnodes
131 @$(MULTI) -n $(B) -l $(INETDIR)/ipnodes;
132 @$(STDHOSTS) -wn $(INETDIR)/ipnodes $(CHKPIPE) | \
133 (awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$1, $$0 }' $(CHKPIPE)) |
134 $(MAKEDBM) $(B) - $(YPDBDIR)/$(DOM)/ipnodes.byaddr;
135 @touch ipnodes.time;
136 @echo "updated ipnodes";
137 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) ipnodes.byname; fi
138 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) ipnodes.byaddr; fi
139 @if [ ! $(NOPUSH) ]; then echo "pushed ipnodes"; fi

141 hosts.time: $(DIR)/hosts
142 @$(MULTI) $(B) -l $(DIR)/hosts;
143 @$(STDHOSTS) -w $(DIR)/hosts $(CHKPIPE) | \
144 (awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$1, $$0 }' $(CHKPIPE)) |
145 $(MAKEDBM) $(B) - $(YPDBDIR)/$(DOM)/hosts.byaddr;
146 @touch hosts.time;
147 @echo "updated hosts";
148 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) hosts.byname; fi
149 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) hosts.byaddr; fi
150 @if [ ! $(NOPUSH) ]; then echo "pushed hosts"; fi

152 ethers.time: $(DIR)/ethers
153 @$(STDETHERS) $(DIR)/ethers $(CHKPIPE) | \
154 (awk 'BEGIN { OFS="\t"; } for (i = 3; i <= NF; i++) print $$i, $$0}' $(CHKPIPE) |
155 $(MAKEDBM) - $(YPDBDIR)/$(DOM)/ethers.byaddr

157 @$(awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$2, $$0 }' \
158 $(DIR)/ethers $(CHKPIPE)) | \
159 $(MAKEDBM) - $(YPDBDIR)/$(DOM)/ethers.byname;
160 @touch ethers.time;
161 @echo "updated ethers";
162 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) ethers.byname; fi
163 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) ethers.byaddr; fi
164 @if [ ! $(NOPUSH) ]; then echo "pushed ethers"; fi

166 networks.time: $(DIR)/networks
167 @$(sed -e '/^#/d' -e s/#.*$$// $(DIR)/networks $(CHKPIPE)) | (awk \
168 '{print $$1, $$0; for (i = 3; i <= NF; i++) print $$i, $$0}' \
169 $(CHKPIPE)) | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/networks.byname;
170 @$(awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$2, $$0 }' \
171 $(DIR)/networks $(CHKPIPE)) | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/networks
172 @touch networks.time;
173 @echo "updated networks";
174 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) networks.byname; fi
175 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) networks.byaddr; fi
176 @if [ ! $(NOPUSH) ]; then echo "pushed networks"; fi

178 services.time: $(DIR)/services
179 @$(awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$2, $$0 }' \
180 $(DIR)/services $(CHKPIPE)) | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/services
181 @$(awk 'BEGIN { OFS="\t"; } \
182 $$1 !~ /^#/ { split($$2, pp, "/"); printf("%s/%s %s\n", $$1, pp[2], $$0); \
183 if (seen[$$1] == "") { \
184 printf("%s %s\n", $$1, $$0); seen[$$1]=$$1; } \
185 for (i = 3; i <= NF && $$i !~ /^#/; i++) { \
186 if (seen[$$i] == "") { \
187 printf("%s %s\n", $$i, $$0); seen[$$i]=$$i; } \
188 printf("%s/%s %s\n", $$i, pp[2], $$0)}}' \
189 $(DIR)/services $(CHKPIPE)) | \

```

```

190 $(MAKEDBM) $(B) - $(YPDBDIR)/$(DOM)/services.byservicename

192 @touch services.time;
193 @echo "updated services";
194 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) services.byname; fi
195 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) services.byservicename; fi
196 @if [ ! $(NOPUSH) ]; then echo "pushed services"; fi

198 rpc.time: $(DIR)/rpc
199 @$(awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$2, $$0 }' \
200 $(DIR)/rpc $(CHKPIPE)) | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/rpc.bynumber;
201 @touch rpc.time;
202 @echo "updated rpc";
203 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) rpc.bynumber; fi
204 @if [ ! $(NOPUSH) ]; then echo "pushed rpc"; fi

206 protocols.time: $(DIR)/protocols
207 @$(awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$2, $$0 }' \
208 $(DIR)/protocols $(CHKPIPE)) | $(MAKEDBM) - \
209 $(YPDBDIR)/$(DOM)/protocols.bynumber;

211 @$(sed -e '/^#/d' -e s/#.*$$// $(DIR)/protocols $(CHKPIPE)) | (awk \
212 '{print $$1, $$0; for (i = 3; i <= NF; i++) print $$i, $$0}' \
213 $(CHKPIPE)) | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/protocols.byname;

215 @touch protocols.time;
216 @echo "updated protocols";
217 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) protocols.byname; fi
218 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) protocols.bynumber; fi
219 @if [ ! $(NOPUSH) ]; then echo "pushed protocols"; fi

221 netgroup.time: $(DIR)/netgroup
222 @$(MAKEDBM) $(DIR)/netgroup $(YPDBDIR)/$(DOM)/netgroup
223 @$(REVNETGROUP) < $(DIR)/netgroup -u $(CHKPIPE) | $(MAKEDBM) - $(YPDBDI
224 @$(REVNETGROUP) < $(DIR)/netgroup -h $(CHKPIPE) | $(MAKEDBM) - $(YPDBDI
225 @touch netgroup.time;
226 @echo "updated netgroup";
227 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) netgroup; fi
228 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) netgroup.byuser; fi
229 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) netgroup.byhost; fi
230 @if [ ! $(NOPUSH) ]; then echo "pushed netgroup"; fi

232 bootparams.time: $(DIR)/bootparams
233 @$(sed -e '/^#/d' -e s/#.*$$// -e 's/[ ]*$$// \
234 -e '\$/s/\$/ /' $(DIR)/bootparams $(CHKPIPE)) \
235 | (awk '/\$/ { printf "%s", $$0 } !/\$/ { print }' $(CHKPIPE)) \
236 | (sed -e 's/[ ]*$$//g' $(CHKPIPE)) \
237 | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/bootparams;
238 @touch bootparams.time;
239 @echo "updated bootparams";
240 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) bootparams; fi
241 @if [ ! $(NOPUSH) ]; then echo "pushed bootparams"; fi

243 aliases.time: $(ALIASES)
244 @cp $(ALIASES) $(YPDBDIR)/$(DOM)/mail.aliases;
245 @/usr/lib/sendmail -bi -oA$(YPDBDIR)/$(DOM)/mail.aliases;
246 $(MKALIAS) $(YPDBDIR)/$(DOM)/mail.aliases $(YPDBDIR)/$(DOM)/mail.byaddr;
247 @rm $(YPDBDIR)/$(DOM)/mail.aliases;
248 @touch aliases.time;
249 @echo "updated aliases";
250 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) mail.aliases; fi
251 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) mail.byaddr; fi
252 @if [ ! $(NOPUSH) ]; then echo "pushed aliases"; fi

254 netmasks.time: $(DIR)/netmasks
255 $(MAKEDBM) $(DIR)/netmasks $(YPDBDIR)/$(DOM)/netmasks.byaddr;

```

```

256 @touch netmasks.time;
257 @echo "updated netmasks";
258 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) netmasks.byaddr; fi
259 @if [ ! $(NOPUSH) ]; then echo "pushed netmasks"; fi

262 publickey.time: $(DIR)/publickey
263 @$(sed "/^#/d" < $(DIR)/publickey $(CHKPIPE)) | $(MAKEDBM) - $(YPBDDIR)/$(
264 @touch publickey.time;
265 @echo "updated publickey";
266 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) publickey.byname; fi
267 @if [ ! $(NOPUSH) ]; then echo "pushed publickey"; fi

269 netid.time: $(PWDIR)/passwd $(DIR)/group $(DIR)/hosts $(DIR)/netid
270 @$ (MKNETID) -q -p $(PWDIR)/passwd -g $(DIR)/group -h $(DIR)/hosts -m $(D
271 @$ (MAKEDBM) .yppjunk $(YPBDDIR)/$(DOM)/netid.byname;
272 @rm -f .yppjunk;
273 @touch netid.time;
274 @echo "updated netid";
275 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) netid.byname; fi
276 @if [ ! $(NOPUSH) ]; then echo "pushed netid"; fi

278 # Old way. Could be restored by PSARC decision.
279 #
280 #passwd.adjunct.time: $(PWDIR)/security/passwd.adjunct
281 # @$(awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { print $$1, $$0 }' $
282 # $(MAKEDBM) -s - $(YPBDDIR)/$(DOM)/passwd.adjunct.byname;
283 # @chmod 600 $(YPBDDIR)/$(DOM)/passwd.adjunct.byname.dir;
284 # @chmod 600 $(YPBDDIR)/$(DOM)/passwd.adjunct.byname.pag;
285 # @touch passwd.adjunct.time
286 # @echo "updated passwd.adjunct";
287 # @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) passwd.adjunct.byname; fi
288 # @if [ ! $(NOPUSH) ]; then echo "pushed passwd.adjunct"; fi

290 passwd.adjunct.time: $(PWDIR)/security/passwd.adjunct $(PWDIR)/shadow
291 -@if [ -f $(PWDIR)/shadow ]; then \
292 (/usr/xpg4/bin/awk 'BEGIN { FS=":"; while (getline < "$(PWDIR)/s
292 (nawk 'BEGIN { FS=":"; while (getline < "$(PWDIR)/shadow" > 0) s
293 else \
294 (awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { print $$1, $
295 $(MAKEDBM) -s - $(YPBDDIR)/$(DOM)/passwd.adjunct.byname; \
296 fi
297 @chmod 600 $(YPBDDIR)/$(DOM)/passwd.adjunct.byname.dir;
298 @chmod 600 $(YPBDDIR)/$(DOM)/passwd.adjunct.byname.pag;
299 @touch passwd.adjunct.time
300 @echo "updated passwd.adjunct";
301 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) passwd.adjunct.byname; fi
302 @if [ ! $(NOPUSH) ]; then echo "pushed passwd.adjunct"; fi

304 group.adjunct.time: $(PWDIR)/security/group.adjunct
305 @$(awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { print $$1, $$0 }' $
306 $(MAKEDBM) -s - $(YPBDDIR)/$(DOM)/group.adjunct.byname;
307 @chmod 600 $(YPBDDIR)/$(DOM)/group.adjunct.byname.dir;
308 @chmod 600 $(YPBDDIR)/$(DOM)/group.adjunct.byname.pag;
309 @touch group.adjunct.time
310 @echo "updated group.adjunct";
311 @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) group.adjunct.byname; fi
312 @if [ ! $(NOPUSH) ]; then echo "pushed group.adjunct"; fi

314 timezone.time: $(DIR)/timezone
315 -@if [ -f $(DIR)/timezone ]; then \
316 sed -e "/^#/d" -e s/#.*$$// $(DIR)/timezone \
317 | awk '{for (i = 2; i<=NF; i++) print $$i, $$0}' \
318 | $(MAKEDBM) - $(YPBDDIR)/$(DOM)/timezone.byname; \
319 touch timezone.time; \
320 echo "updated timezone"; \

```

```

321 if [ ! $(NOPUSH) ]; then \
322 $(YPPUSH) timezone.byname; \
323 echo "pushed timezone"; \
324 else \
325 : ; \
326 fi \
327 else \
328 echo "couldn't find $(DIR)/timezone"; \
329 fi

331 auto.master.time: $(DIR)/auto_master
332 -@if [ -f $(DIR)/auto_master ]; then \
333 sed -e "/^#/d" -e s/#.*$$// $(DIR)/auto_master \
334 | $(MAKEDBM) - $(YPBDDIR)/$(DOM)/auto.master; \
335 touch auto.master.time; \
336 echo "updated auto.master"; \
337 if [ ! $(NOPUSH) ]; then \
338 $(YPPUSH) auto.master; \
339 echo "pushed auto.master"; \
340 else \
341 : ; \
342 fi \
343 else \
344 echo "couldn't find $(DIR)/auto_master"; \
345 fi

347 auto.home.time: $(DIR)/auto_home
348 -@if [ -f $(DIR)/auto_home ]; then \
349 sed -e "/^#/d" -e s/#.*$$// $(DIR)/auto_home \
350 | $(MAKEDBM) - $(YPBDDIR)/$(DOM)/auto.home; \
351 touch auto.home.time; \
352 echo "updated auto.home"; \
353 if [ ! $(NOPUSH) ]; then \
354 $(YPPUSH) auto.home; \
355 echo "pushed auto.home"; \
356 else \
357 : ; \
358 fi \
359 else \
360 echo "couldn't find $(DIR)/auto_home"; \
361 fi

364 auth.attr.time: $(RBACDIR)/auth_attr
365 -@if [ -f $(RBACDIR)/auth_attr ]; then \
366 sed -e "/^#/d" -e s/#.*$$// $(RBACDIR)/auth_attr \
367 | sed -e '/\|$$/{:l' -e 'N;s/\|\\n//;t h' -e ':h' \
368 -e 's/\|$$/\|/;t l' -e } \
369 | (/usr/xpg4/bin/awk 'BEGIN { FS=":"; OFS=":" } /^[a-zA-Z0-9_]/
369 (nawk 'BEGIN { FS=":"; OFS=":" } /^[a-zA-Z0-9_]/ \
370 | printf "%s:%s\n", $$1, $$0 }' $(CHKPIPE)) \
371 | $(MAKEDBM) -s ":-E - $(YPBDDIR)/$(DOM)/auth_attr; \
372 touch auth.attr.time; \
373 echo "updated auth_attr"; \
374 if [ ! $(NOPUSH) ]; then \
375 $(YPPUSH) auth_attr; \
376 echo "pushed auth_attr"; \
377 else \
378 : ; \
379 fi \
380 else \
381 echo "couldn't find $(RBACDIR)/auth_attr"; \
382 fi

384 exec.attr.time: $(RBACDIR)/exec_attr
385 -@if [ -f $(RBACDIR)/exec_attr ]; then \

```

```

386     sed -e "/^#/d" -e s/#.*$$// $(RBACDIR)/exec_attr \
387     |sed -e '/\$\${:l' -e 'Nis/\n//;t h' -e 'h' \
388     -e 's/\$\$/\$/;t l' -e } \
389     | (/usr/xpg4/bin/awk 'BEGIN { FS=";"; OFS=":" } /^[a-zA-Z0-9_]/
389     | (nawk 'BEGIN { FS=";"; OFS=":" } /^[a-zA-Z0-9_]/ \
390     {printf "%s:%s\n", $$1, $$2, $$6, $$0 }' $(CHKPIPE)) \
391     | $(MAKEDBM) -S ":" -E -D 2 - $(YPDBDIR)/$(DOM)/exec_attr; \
392     touch exec.attr.time; \
393     echo "updated exec_attr"; \
394     if [ ! $(NOPUSH) ]; then \
395         $(YPPUSH) exec_attr; \
396         echo "pushed exec_attr"; \
397     else \
398         ; ; \
399     fi \
400 else \
401     echo "couldn't find $(RBACDIR)/exec_attr"; \
402 fi

404 prof.attr.time: $(RBACDIR)/prof_attr
405     -@if [ -f $(RBACDIR)/prof_attr ]; then \
406     sed -e "/^#/d" -e s/#.*$$// $(RBACDIR)/prof_attr \
407     |sed -e '/\$\${:l' -e 'Nis/\n//;t h' -e 'h' \
408     -e 's/\$\$/\$/;t l' -e } \
409     | (/usr/xpg4/bin/awk 'BEGIN { FS=";"; OFS=":" } /^[a-zA-Z0-9_]/
409     | (nawk 'BEGIN { FS=";"; OFS=":" } /^[a-zA-Z0-9_]/ \
410     {printf "%s:%s\n", $$1, $$0 }' $(CHKPIPE)) \
411     | $(MAKEDBM) -S ":" -E - $(YPDBDIR)/$(DOM)/prof_attr; \
412     touch prof.attr.time; \
413     echo "updated prof_attr"; \
414     if [ ! $(NOPUSH) ]; then \
415         $(YPPUSH) prof_attr; \
416         echo "pushed prof_attr"; \
417     else \
418         ; ; \
419     fi \
420 else \
421     echo "couldn't find $(RBACDIR)/prof_attr"; \
422 fi

424 user.attr.time: $(DIR)/user_attr
425     -@if [ -f $(DIR)/user_attr ]; then \
426     sed -e "/^#/d" -e s/#.*$$// $(DIR)/user_attr \
427     |sed -e '/\$\${:l' -e 'Nis/\n//;t h' -e 'h' \
428     -e 's/\$\$/\$/;t l' -e } \
429     | (/usr/xpg4/bin/awk 'BEGIN { FS=";"; OFS=":" } /^[a-zA-Z0-9_]/
429     | (nawk 'BEGIN { FS=";"; OFS=":" } /^[a-zA-Z0-9_]/ \
430     {printf "%s:%s\n", $$1, $$0 }' $(CHKPIPE)) \
431     | $(MAKEDBM) -S ":" -E - $(YPDBDIR)/$(DOM)/user_attr; \
432     touch user.attr.time; \
433     echo "updated user_attr"; \
434     if [ ! $(NOPUSH) ]; then \
435         $(YPPUSH) user_attr; \
436         echo "pushed user_attr"; \
437     else \
438         ; ; \
439     fi \
440 else \
441     echo "couldn't find $(DIR)/user_attr"; \
442 fi

444 ageing.time: $(PWDIR)/shadow
445     -@if [ -f $(PWDIR)/shadow ]; then \
446     (awk 'BEGIN {FS=";"; OFS=":"} $$1 !~ /^#/ {printf "%s\t%s:%s:%s:
447     touch ageing.time; \
448     echo "updated ageing"; \

```

```

449     else \
450     echo "couldn't find $(PWDIR)/shadow"; \
451     fi

453 passwd: passwd.time
454 group: group.time
455 project: project.time
456 hosts: hosts.time
457 ipnodes: ipnodes.time
458 ethers: ethers.time
459 networks: networks.time
460 rpc: rpc.time
461 services: services.time
462 protocols: protocols.time
463 netgroup: netgroup.time
464 bootparams: bootparams.time
465 aliases: aliases.time
466 publickey: publickey.time
467 netid: netid.time
468 passwd.adjunct: passwd.adjunct.time
469 group.adjunct: group.adjunct.time
470 netmasks: netmasks.time
471 timezone: timezone.time
472 auto.master: auto.master.time
473 auto.home: auto.home.time
474 auth.attr:auth.attr.time
475 exec.attr:exec.attr.time
476 prof.attr:prof.attr.time
477 user.attr:user.attr.time
478 $(DIR)/netid:
479 $(DIR)/timezone:
480 $(DIR)/auto_master:
481 $(DIR)/auto_home:
482 $(PWDIR)/shadow:
483 $(DIR)/auth_attr:
484 $(DIR)/exec_attr:
485 $(DIR)/prof_attr:
486 $(DIR)/user_attr:
487 ageing: ageing.time

```

new/usr/src/cmd/ypcmd/ypstart.sh

1

```
*****
3190 Sun May 26 18:22:56 2013
new/usr/src/cmd/ypcmd/ypstart.sh
XXX Remove nawk(1)
*****
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright 2006 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # ident "%Z%M% %I% %E% SMI"
27 #
28 # Enable appropriate NIS daemons based on the current configuration.
29
30 enable () {
31     /usr/sbin/svccadm enable -t $1
32     [ $? = 0 ] || echo "ypstart: unable to enable $1"
33
34     if [ "`/usr/bin/svcprop -p restarter/state $1`" = "maintenance" ]; then
35         echo "ypstart: unable to enable $1; in maintenance"
36     fi
37 }
38
39 domain='domainname'
40 if [ -z "$domain" ]; then
41     echo "ERROR: Default domain is not defined. \c"
42     echo "Use \"domainname\" to set the domain."
43     exit 1
44 fi
45
46 echo "starting NIS (YP server) services:\c"
47
48 zone='/sbin/zonename'
49
50 if [ -d /var/yp/$domain ]; then
51     state='/usr/bin/svcprop -p restarter/state network/nis/server:default'
52     [ "$state" = "disabled" ] && if [ -n "`pgrep -z $zone ypserv`" ]; then
53         echo "ypstart: ypserv already running?"
54     fi
55
56     enable svc:/network/nis/server:default && echo " ypserv\c"
57
58     YP_SERVER=TRUE # remember we're a server for later
```

new/usr/src/cmd/ypcmd/ypstart.sh

2

```
60 # check to see if we are the master
61 if [ -f /var/yp/NISLDAPmapping ]; then
62     passwdfile=/var/yp/$domain/LDAP_passwd.byname
63 else
64     passwdfile=/var/yp/$domain/passwd.byname
65 fi
66 master='/usr/sbin/makedbm -u $passwdfile | grep YP_MASTER_NAME \
67 | /usr/xpg4/bin/awk '{ print tolower($2) }' \
68 | nawk '{ print tolower($2) }'`
69 fi
70 # Enabling the YP client is not strictly necessary, but it is
71 # traditional.
72 state='/usr/bin/svcprop -p restarter/state network/nis/client:default'
73
74 [ "$state" = "disabled" ] && if [ -n "`pgrep -z $zone ypbind`" ]; then
75     echo "ypstart: ypbind already running?"
76 fi
77
78 enable svc:/network/nis/client:default && echo " ypbind\c"
79
80 # do a ypwhich to force ypbind to get bound
81 ypwhich > /dev/null 2>&1
82
83 if [ "$YP_SERVER" = TRUE ]; then
84     # Are we the master server? If so, start the
85     # ypxfrd, rpc.yppasswdd and rpc.yppupdated daemons.
86     hostname='uname -n | tr '[A-Z]' '[a-z]`'
87
88     if [ "$master" = "$hostname" ]; then
89         enable svc:/network/nis/xfr:default && echo " ypxfrd\c"
90         enable svc:/network/nis/passwd:default &&
91             echo " rpc.yppasswdd\c"
92
93         if [ ! -f /var/yp/NISLDAPmapping -a -f /var/yp/updaters ]; then
94             enable svc:/network/nis/update:default &&
95                 echo " rpc.yppupdated\c"
96         fi
97     fi
98 fi
99
100 # As this operation is likely configuration changing, restart the
101 # name-services milestone (such that configuration-sensitive services
102 # are in turn restarted).
103 /usr/sbin/svccadm restart milestone/name-services
104
105 echo " done."
```

new/usr/src/cmd/zic/do_pofile.sh

1

```
*****
1042 Sun May 26 18:22:57 2013
new/usr/src/cmd/zic/do_pofile.sh
XXX Remove awk(1)
*****
1 #! /bin/sh
2 #
3 # Copyright (c) 2001 by Sun Microsystems, Inc.
4 # All rights reserved.
5 #
6 # ident "%Z%M% %I% %E% SMI"
7 #
8 # Create messages file for zone_sun.tab, country.tab, continent.tab
9 #
10 AWK=/usr/xpg4/bin/awk
11 AWK=/usr/bin/awk
11 ECHO=/usr/bin/echo
13 ZONE_SUN_FILE=zone_sun.tab
14 COUNTRY_FILE=country.tab
15 CONTINENT_FILE=continent.tab
17 DOMAIN=SUNW_OST_ZONEINFO
19 $ECHO "domain \"\$DOMAIN\""
21 $ECHO "#"
22 $ECHO "# These files are located in usr/src/cmd/zic."
23 $ECHO "#"
25 $ECHO "#"
26 $ECHO "# continent.tab"
27 $ECHO "#"
29 # Get 2nd column of continent.tab file
30 $AWK -F'\t' \
31 ,
32 /^#/ { next }
33 {
34     printf "msgid \"%s\"\n", $2
35     printf "msgstr \"\"\n"
36 }
37 ' < $CONTINENT_FILE
39 $ECHO "#"
40 $ECHO "# country.tab"
41 $ECHO "#"
43 # Get 2nd column of country.tab file
44 $AWK -F'\t' \
45 ,
46 /^#/ { next }
47 {
48     printf "msgid \"%s\"\n", $2
49     printf "msgstr \"\"\n"
50 }
51 ' < $COUNTRY_FILE
54 $ECHO "#"
55 $ECHO "# zone.tab"
56 $ECHO "#"
58 # Get 5th column (if it exists) of zone_sun.tab file
59 $AWK -F'\t' \
```

new/usr/src/cmd/zic/do_pofile.sh

2

```
60 '
61 /^#/ { next }
62 {
63     if (NF > 4) {
64         printf "msgid \"%s\"\n", $5
65         printf "msgstr \"\"\n"
66     }
67 }
68 ' < $ZONE_SUN_FILE
```

new/usr/src/cmd/zic/tzselect.ksh

1

```
*****
10405 Sun May 26 18:22:57 2013
new/usr/src/cmd/zic/tzselect.ksh
XXX Remove awk(1)
*****
1 #! /usr/bin/ksh
2 #
3 # Copyright 2005 Sun Microsystems, Inc. All rights reserved.
4 # Use is subject to license terms.
5 #
6 # ident "%Z%M% %I% %E% SMI"
7 #
8 # '@(#)tzselect.ksh 1.8'

10 # Ask the user about the time zone, and output the resulting TZ value to stdout.
11 # Interact with the user via stderr and stdin.

13 # Contributed by Paul Eggert

15 # Porting notes:
16 #
17 # This script requires several features of the Korn shell.
18 # If your host lacks the Korn shell,
19 # you can use either of the following free programs instead:
20 #
21 # <a href=ftp://ftp.gnu.org/pub/gnu/>
22 # Bourne-Again shell (bash)
23 # </a>
24 #
25 # <a href=ftp://ftp.cs.mun.ca/pub/pdksh/pdksh.tar.gz>
26 # Public domain ksh
27 # </a>
28 #
29 # This script also uses several features of modern awk programs.
30 # If your host lacks awk, or has an old awk that does not conform to POSIX.2,
31 # you can use either of the following free programs instead:
32 #
33 # <a href=ftp://ftp.gnu.org/pub/gnu/>
34 # GNU awk (gawk)
35 # </a>
36 #
37 # <a href=ftp://ftp.whidbey.net/pub/brennan/>
38 # mawk
39 # </a>

41 AWK=/usr/xpg4/bin/awk
41 AWK=/usr/bin/nawk
42 GREP=/usr/bin/grep
43 EXPR=/usr/bin/expr
44 LOCALE=/usr/bin/locale
45 SORT=/usr/bin/sort
46 PRINTF=/usr/bin/printf
47 DATE=/usr/bin/date
48 GETTEXT=/usr/bin/gettext

50 TZDIR=/usr/share/lib/zoneinfo

52 # Messages
53 ERR_NO_SETUP="%s: time zone files are not set up correctly"
54 INFO_LOCATION="Please identify a location so that time zone rules \
55 can be set correctly."
56 INFO_SELECT_CONT="Please select a continent or ocean."
57 INFO_POSIX="none - I want to specify the time zone using the POSIX \
58 TZ format."
59 WARN_ENTER_NUM="Please enter a number in range."
60 INFO_ENTER_POSIX="Please enter the desired value of the TZ environment \
```

new/usr/src/cmd/zic/tzselect.ksh

2

```
61 variable."
62 INFO_POSIX_EX="For example, GST-10 is a zone named GST that is 10 hours \
63 ahead (east) of UTC."
64 ERR_INV_POSIX="\`%s\` is not a conforming POSIX time zone string."
65 INFO_SELECT_CNTRY="Please select a country or region."
66 INFO_SELECT_TZ="Please select one of the following time zone regions."
67 INFO_EXTRAL1="Local time is now: %s"
68 INFO_EXTRAL2="Universal Time is now: %s"
69 INFO_INFO="The following information has been given:"
70 INFO_TZ="Therefore TZ='`%s`' will be used."
71 INFO_OK="Is the above information OK?"
72 INFO_YES="Yes"
73 INFO_NO="No"
74 WARN_ENTER_YORN="Please enter 1 for Yes, or 2 for No."
75 INFO_FINE="Here is the TZ value again, this time on standard output:"

77 # I18n support
78 TEXTDOMAINDIR=/usr/lib/locale; export TEXTDOMAINDIR
79 TEXTDOMAIN=SUNW_OST_OSCMD; export TEXTDOMAIN
80 DOMAIN2=SUNW_OST_ZONEINFO

82 # Make sure the tables are readable.
83 TZ_COUNTRY_TABLE=$TZDIR/tab/country.tab
84 TZ_ZONE_TABLE=$TZDIR/tab/zone_sun.tab
85 for f in $TZ_COUNTRY_TABLE $TZ_ZONE_TABLE
86 do
87     <$f || {
88         $PRINTF >&2 "`$GETTEXT "$ERR_NO_SETUP"`\n" $0
89         exit 1
90     }
91 done

93 newline='
94 '
95 IFS=$newline

97 # For C locale, don't need to call gettext(1)
98 loc_messages=$LOCALE | $GREP LC_MESSAGES | $AWK -F=" " '{print $2}'
99 if [ "$loc_messages" = "\C" -o "$loc_messages" = "C" ]; then
100     is_C=1
101 else
102     is_C=0
103 fi

105 iafrica='$GETTEXT $DOMAIN2 Africa'
106 iamericas='$GETTEXT $DOMAIN2 Americas'
107 iantarctica='$GETTEXT $DOMAIN2 Antarctica'
108 iarcticoc ocean='$GETTEXT $DOMAIN2 "Arctic Ocean"'
109 iasia='$GETTEXT $DOMAIN2 Asia'
110 iatlanticoc ocean='$GETTEXT $DOMAIN2 "Atlantic Ocean"'
111 iaustralia='$GETTEXT $DOMAIN2 Australia'
112 ieurop e='$GETTEXT $DOMAIN2 Europe'
113 ipacificoc ocean='$GETTEXT $DOMAIN2 "Pacific Ocean"'
114 iindianocean='$GETTEXT $DOMAIN2 "Indian Ocean"'
115 none='$GETTEXT "$INFO_POSIX"'

117 # Begin the main loop. We come back here if the user wants to retry.
118 while
119     $PRINTF >&2 "`$GETTEXT "$INFO_LOCATION"`\n"

121     continent=
122     country=
123     region=

125     # Ask the user for continent or ocean.
126     $PRINTF >&2 "`$GETTEXT "$INFO_SELECT_CONT"`\n"
```



```

128 select continent in \
129     $iafrica \
130     $iamerica \
131     $iantarctica \
132     $iarcticocean \
133     $iasia \
134     $iatlanticocean \
135     $iaustralia \
136     $ieurope \
137     $iindianocean \
138     $ipacificocean \
139     $none

141 do
142 case $continent in
143     '')
144     $PRINTF >&2 "$GETTEXT "$WARN_ENTER_NUM"\n";;

146     ?*)
147     case $continent in
148     $iafrica) continent=Africa;;
149     $iamerica) continent=America;;
150     $iantarctica) continent=Antarctica;;
151     $iarcticocean) continent=Arctic;;
152     $iasia) continent=Asia;;
153     $iatlanticocean) continent=Atlantic;;
154     $iaustralia) continent=Australia;;
155     $ieurope) continent=Europe;;
156     $iindianocean) continent=Indian;;
157     $ipacificocean) continent=Pacific;;
158     $none) continent=none;;
159     esac
160     break
161     esac
162 done
163 case $continent in
164     '')
165     exit 1;;
166     none)
167     # Ask the user for a POSIX TZ string. Check that it conforms.
168     while
169     $PRINTF >&2 "$GETTEXT "$INFO_ENTER_POSIX"\n"
170     $PRINTF >&2 "$GETTEXT "$INFO_POSIX_EX"\n"

172     read TZ
173     env LC_ALL=C $AWK -v TZ="$TZ" 'BEGIN {
174     tzname = "[^+,0-9][^+,0-9][^+,0-9]?"
175     time = "[0-2]?[0-9](:[0-5][0-9](:[0-5][0-9])?)?"
176     offset = "[+]??" time
177     date = "(J?[0-9]+|M[0-9]+\.[0-9]+\.[0-9]+)"
178     datetime = "," date ("/" time)?"
179     tzpattern = "^(:.*)? tzname offset (" tzname \
180     "(" offset")?(" datetime datetime")?)?"
181     if (TZ ~ tzpattern) exit 1
182     exit 0
183     }'
184     do
185     $PRINTF >&2 "$GETTEXT "$ERR_INV_POSIX"\n" $TZ

187     done
188     TZ_for_date=$TZ;;
189     *)
190     # Get list of names of countries in the continent or ocean.
191     countries=$(($AWK -F'\t' \
192     -v continent="$continent" \

```

```

193     -v TZ_COUNTRY_TABLE="$TZ_COUNTRY_TABLE" \
194     ,
195     /^#/ { next }
196     $3 ~ {"^" continent "/" } {
197     if (!cc_seen[$1]++) cc_list[++ccs] = $1
198     }
199     END {
200     while (getline <TZ_COUNTRY_TABLE) {
201     if ($0 !~ /^#/ ) cc_name[$1] = $2
202     }
203     for (i = 1; i <= ccs; i++) {
204     country = cc_list[i]
205     if (cc_name[country]) {
206     country = cc_name[country]
207     }
208     print country
209     }
210     }
211     ' <$TZ_ZONE_TABLE | $SORT -f)

213 # il8n country names
214 c=0
215 set -A icountry
216 for country in $countries
217 do
218     if [ $is_C -eq 1 ]; then
219     icountry[c]=$country
220     else
221     icountry[c]='${GETTEXT} ${DOMAIN2} $country'
222     fi
223     ocountry[c]="$country"
224     c=$(( $c + 1 ))
225 done
226 maxnum=$c

228 # If there's more than one country, ask the user which one.
229 case $countries in
230     *"$newline"*)
231     $PRINTF >&2 "$GETTEXT "$INFO_SELECT_CNTRY"\n"
232     select xcountry in ${icountry[*]}
233     do
234     case $xcountry in
235     '')
236     $PRINTF >&2 "$GETTEXT "$WARN_ENTER_NUM"\n"
237     ;;
238     ?*)
239     c=0
240     while true; do
241     if [ "$xcountry" = "${icountry[$c]} " ];
242     then
243     country="${ocountry[$c]}"
244     break
245     fi
246     if [ $c -lt $maxnum ]; then
247     c=$(( $c + 1 ))
248     else
249     break
250     fi
251     done
252     break
253     esac
254     done

255     case $xcountry in
256     '') exit 1
257     esac;;
258     *)

```

```

259         country=$countries
260         xcountry=$countries
261     esac

264     # Get list of names of time zone rule regions in the country.
265     regions=$(SAWK -F'\t' \
266         -v country="$country" \
267         -v TZ_COUNTRY_TABLE="$TZ_COUNTRY_TABLE" \
268         ,
269         BEGIN {
270             cc = country
271             while (getline <TZ_COUNTRY_TABLE) {
272                 if ($0 !~ /^#/ && country == $2) {
273                     cc = $1
274                     break
275                 }
276             }
277             $1 == cc { print $5 }
278         } <$TZ_ZONE_TABLE)
279

281     # Il8n region names
282     c=0
283     set -A iregion
284     for region in $regions
285     do
286         if [ $is_C -eq 1 ]; then
287             iregion[c]=$region
288         else
289             iregion[c]='${GETTEXT} ${DOMAIN2} $region'
290         fi
291         oregion[c]="$region"
292         c=$(( $c + 1 ))
293     done
294     maxnum=$c

296     # If there's more than one region, ask the user which one.
297     case $regions in
298     *"$newline"*)
299         $PRINTF >&2 "\${GETTEXT} "$INFO_SELECT_TZ""\n"

301         select xregion in ${iregion[*]}
302         do
303             case $xregion in
304             '')
305                 $PRINTF >&2 "\${GETTEXT} "$WARN_ENTER_NUM""\n"
306                 ;;
307             ?*) c=0
308                 while true; do
309                     if [ "$xregion" = "${iregion[$c]}" ]; then
310                         region="${oregion[$c]}"
311                         break
312                     fi
313                     if [ $c -lt $maxnum ]; then
314                         c=$(( $c + 1 ))
315                     else
316                         break
317                     fi
318                 done
319             done
320             break
321         esac
322     done

324     case $region in

```

```

325         '' ) exit 1
326         esac;;
327     *)
328         region=$regions
329         xregion=$regions
330     esac

332     # Determine TZ from country and region.
333     TZ=$(SAWK -F'\t' \
334         -v country="$country" \
335         -v region="$region" \
336         -v TZ_COUNTRY_TABLE="$TZ_COUNTRY_TABLE" \
337         ,
338         BEGIN {
339             cc = country
340             while (getline <TZ_COUNTRY_TABLE) {
341                 if ($0 !~ /^#/ && country == $2) {
342                     cc = $1
343                     break
344                 }
345             }
346             $1 == cc && $5 == region {
347                 # Check if tzname mapped to
348                 # backward compatible tzname
349                 if ($4 == "-") {
350                     print $3
351                 } else {
352                     print $4
353                 }
354             }
355         } <$TZ_ZONE_TABLE)
356
357     # Make sure the corresponding zoneinfo file exists.
358     TZ_for_date=$TZDIR/$TZ
359     <$TZ_for_date || {
360         $PRINTF >&2 "\${GETTEXT} "$ERR_NO_SETUP""\n" $0
361         exit 1
362     }
363     # Absolute path TZ's not supported
364     TZ_for_date=$TZ
365     esac

367     # Use the proposed TZ to output the current date relative to UTC.
368     # Loop until they agree in seconds.
369     # Give up after 8 unsuccessful tries.

374     extra_info1=
375     extra_info2=
376     for i in 1 2 3 4 5 6 7 8
377     do
378         TZdate=$(LANG=C TZ="$TZ_for_date" $DATE)
379         UTdate=$(LANG=C TZ=UTC0 $DATE)
380         TZsec=$(($EXPR "$TZdate" : '.*:([0-5][0-9]\)'))
381         UTsec=$(($EXPR "$UTdate" : '.*:([0-5][0-9]\)'))
382         case $TZsec in
383         $UTsec)
384             extra_info1=$(($PRINTF "\${GETTEXT} "$INFO_EXTRA1"" \
385                 "$TZdate")
386             extra_info2=$(($PRINTF "\${GETTEXT} "$INFO_EXTRA2"" \
387                 "$UTdate")
388             break
389         esac
390     done

```

```
393      # Output TZ info and ask the user to confirm.
395      $PRINTF >&2 "\n"
396      $PRINTF >&2 "$INFO_INFO"\n"
397      $PRINTF >&2 "\n"

399      case $country+$region in
400      ?*+?) $PRINTF >&2 " $xcountry$newline $xregion\n";;
401      ?*+) $PRINTF >&2 " $xcountry\n";;
402      +)   $PRINTF >&2 " TZ='$TZ'\n"
403      esac
404      $PRINTF >&2 "\n"
405      $PRINTF >&2 "$INFO_TZ"\n" "$TZ"
406      $PRINTF >&2 "$extra_info1\n"
407      $PRINTF >&2 "$extra_info2\n"
408      $PRINTF >&2 "$INFO_OK"\n"

410      ok=
411      # select ok in Yes No
412      Yes="$GETTEXT "$INFO_YES""
413      No="$GETTEXT "$INFO_NO""
414      select ok in $Yes $No
415      do
416          case $ok in
417          '') $PRINTF >&2 "$WARN_ENTER_YORN"\n"
418              ;
419              ?*) break
420          esac
421      done
422      case $ok in
423      '') exit 1;
424      $Yes) break
425      esac
426  do :
427 done

430 $PRINTF >&2 "$GETTEXT "$INFO_FINE"\n"

432 $PRINTF "%s\n" "$TZ"
```

```

*****
4600 Sun May 26 18:22:58 2013
new/usr/src/cmd/zoneadm/svc-zones
XXX Remove awk(1)
*****
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.

25 . /lib/svc/share/smf_include.sh

27 #
28 # Return a list of running, non-global zones for which a shutdown via
29 # "/sbin/init 0" may work (typically only Solaris zones.)
30 #
31 shutdown_zones()
32 {
33     zoneadm list -p | /usr/xpg4/bin/awk -F: '{
34     zoneadm list -p | awk -F: '{
35         if ($2 != "global") {
36             print $2
37         }
38     }'
39 }

40 [ ! -x /usr/sbin/zoneadm ] && exit 0 # SUNWzoneu not installed

42 if [ -z "$SMF_FMRI" ]; then
43     echo "this script can only be invoked by smf(5)"
44     exit $SMF_EXIT_ERR_NOSMF
45 fi

47 # Make sure working directory is / to prevent unmounting problems.
48 cd /
49 PATH=/usr/sbin:/usr/bin; export PATH

51 case "$1" in
52 'start')
53     egrep -vs '^#|^global:' /etc/zones/index || exit 0 # no local zones

55     #
56     # Boot the installed zones for which the "autoboot" zone property is
57     # set and invoke the sysboot hook for all other installed zones.
58     #
59     ZONES=""
60     for zone in `zoneadm list -pi | /usr/xpg4/bin/awk -F: '{

```

```

60     for zone in `zoneadm list -pi | awk -F: '{
61         if ($3 == "installed") {
62             print $2
63         }
64     }'; do
65     zonecfg -z $zone info autoboot | grep "true" >/dev/null 2>&1
66     if [ $? -eq 0 ]; then
67         [ -z "$ZONES" ] && echo "Booting zones:\c"
68         ZONES=yes
69         echo " $zone\c"
70         #
71         # zoneadm puts itself into its own contract so
72         # this service will lose sight of it. We don't
73         # support restart so it is OK for zoneadm to
74         # be in an orphaned contract.
75         #
76         zoneadm -z $zone boot &
77     else
78         zoneadm -z $zone sysboot &
79     fi
80 done

82 #
83 # Wait for all zoneadm processes to finish before allowing the
84 # start method to exit.
85 #
86 wait
87 [ -n "$ZONES" ] && echo .
88 ;

90 'stop')
91     egrep -vs '^#|^global:' /etc/zones/index || exit 0 # no local zones
92     [ "`zoneadm list`" = "global" ] && exit 0 # no zones running

94     SVC_TIMEOUT='svcprop -p stop/timeout_seconds $SMF_FMRI'

96     #
97     # First, try shutting down any running zones for which an "init 0" may
98     # work.
99     #
100     MAXSHUT='expr 3 \* $SVC_TIMEOUT \/ 4' # 3/4 of time to zone shutdown
101     MAXHALT='expr $SVC_TIMEOUT \/ 4' # rest of time goes to halt

103     zonelist='shutdown_zones'

105     if [ -n "$zonelist" ]; then
106         SHUTDOWN=0
107         echo "Shutting down running zones (for up to $MAXSHUT \
108             \"seconds):\c"

110         for zone in $zonelist; do
111             echo " $zone\c"
112             zlogin -S $zone /sbin/init 0 < /dev/null >&0 2>&0 &
113             SHUTDOWN=1
114         done

116         [ $SHUTDOWN -eq 1 ] && echo "."

118         # Allow time for zones to shutdown cleanly

120         while [ $MAXSHUT -gt 0 -a "`shutdown_zones`" != "" ]; do
121             MAXSHUT='expr $MAXSHUT - 1'
122             sleep 1 # wait a bit longer
123         done
124     fi

```

```
126 #
127 # Second, try halting any non-global zones still running
128 #
129 WAITPIDS=""
130 for zone in `zoneadm list`; do
131     if [ "$zone" != "global" ]; then
132         [ -z "$WAITPIDS" ] &&
133             echo "Zones failed to shutdown; trying to halt " \
134                 "(for up to $MAXHALT seconds):\c"
135             echo " $zone\c"
136             zoneadm -z $zone halt &
137             WAITPIDS="$WAITPIDS $!"
138         fi
139     done
140     [ ! -z "$WAITPIDS" ] && echo .
141
142 # Wait for the 'zoneadm halt' commands to complete. We will let this
143 # run forever, since the restart daemon will eventually kill us off
144 # anyway if the halts do not complete after a certain period of time.
145 wait $WAITPIDS
146
147 # If the halts complete but a zone is still not shutdown, it might
148 # be in a state like 'shutting_down' or 'down'. So we give it some
149 # time to come all the way down.
150
151 while [ $MAXHALT -gt 0 -a "`zoneadm list`" != "global" ]; do
152     MAXHALT=`expr $MAXHALT - 1`
153     sleep 1 # wait a bit longer
154 done
155
156 # If there are any remaining file systems in zones, try to unmount them.
157 umountall -Z
158
159 #
160 # Report on zones which failed to shutdown.
161 #
162 for zone in `zoneadm list`; do
163     if [ "$zone" != "global" ]; then
164         echo "Zone '$zone' failed to halt."
165     fi
166 done
167 [ "`zoneadm list`" != "global" ] && exit 1 # zones still running
168 ;;
169
170 *)
171     echo "Usage: $0 { start | stop }"
172     exit 1
173 ;;
174 esac
175 exit 0
```

```

*****
4468 Sun May 26 18:22:58 2013
new/usr/src/common/mapfiles/gen/Makefile
XXX Remove awk(1)
*****
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 2007 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # ident "%Z%M% %I% %E% SMI"
27 #
28 #
29 include $(SRC)/Makefile.master
30 #
31 #
32 #
33 $(BUILD64)FILES += \
34 $(MACH64)_cc_map.noexeglobs \
35 $(MACH64)_gcc_map.noexeglobs
36 #
37 SYMS1= syms.1
38 SYMS2= syms.2
39 MAIN1= main.1
40 MAIN2= main.2
41 #
42 TEMPLATE1= map.noexeglobs.1.template
43 TEMPLATE2= map.noexeglobs.2.template
44 #
45 all install: $(FILES)
46 #
47 lint:
48 #
49 clean:
50 $(RM) $(SYMS1) $(SYMS2) $(MAIN1) $(MAIN2)
51 #
52 clobber:
53 clean
54 $(RM) $(FILES)
55 #
56 # A number of dynamic executables have their own definitions of interfaces that
57 # exist in system libraries. To prevent name-space pollution it is desirable
58 # to demote the interfaces within these executable to locals. However, various
59 # symbols defined by the compiler drivers crt/values files need to remain
60 # global in any dynamic object that includes these files. These symbols

```

```

60 # interpose on symbols in libc, or provide call backs for other system
61 # libraries. The various compiler drivers (cc and gcc), and the crt/values
62 # files that these drivers are configured to include, differ between the
63 # various compilation environments (platform, 32/64-bit). Therefore, the
64 # only means of creating a mapfile to demote symbols is to dynamically generate
65 # the mapfile for a specific compilation environment.
66 #
67 # Here, template mapfiles are used to generate a number of compilation specific
68 # mapfiles. These mapfiles are referenced by components of the build through
69 # the MAPFILE.NGB macro defined in Makefile.master. These dynamically created
70 # mapfiles are not delivered into the $ROOT area, and therefore are not
71 # delivered as packaged components of the OSNet.
72 #
73 $(MACH)_cc_map.noexeglobs := LINK = $(LINK.c)
74 $(MACH64)_cc_map.noexeglobs := LINK = $(LINK64.c)
75 #
76 $(MACH)_gcc_map.noexeglobs := CC = $(ONBLD_TOOLS)/bin/$(MACH)/cw -gcc
77 $(MACH)_gcc_map.noexeglobs := LINK = $(LINK.c)
78 $(MACH64)_gcc_map.noexeglobs := CC = $(ONBLD_TOOLS)/bin/$(MACH)/cw -gcc
79 $(MACH64)_gcc_map.noexeglobs := LINK = $(LINK64.c)
80 #
81 # This generic target creates two dynamic executables from an empty "main"
82 # program. These objects are not executed, but are analyzed to determine the
83 # global symbols each provides.
84 #
85 # The first executable demotes a family of known interfaces to local and allows
86 # all other symbol definitions to remain global. This executables provides the
87 # base for discovering all symbol definitions provided by the various
88 # compilation environments. The second executable demotes all symbols to
89 # locals. Within both executables, some symbols remain globals (_end, _etext,
90 # etc.) as the link-editor has special knowledge of these symbols and their
91 # expected visibility requirements. By inspecting the deferences between the
92 # global symbols within the two executables, a mapfile can be generated to
93 # ensure the symbols defined by the compilation environments files remain
94 # global.
95 #
96 %map.noexeglobs:main.c $(TEMPLATE1) $(TEMPLATE2)
97 $(LINK) -o $(MAIN1) -M$(TEMPLATE1) main.c
98 $(ELFDUMP) -s -N.dynsym $(MAIN1) | $(EGREP) "WEAK|GLOB" | \
99 $(GREP) -v UNDEF | $(AWK) '{print $$9}' | $(SORT) > $(SYMS1)
100 $(LINK) -o $(MAIN2) -M$(TEMPLATE2) main.c
101 $(ELFDUMP) -s -N.dynsym $(MAIN2) | $(EGREP) "WEAK|GLOB" | \
102 $(GREP) -v UNDEF | $(AWK) '{print $$9}' | $(SORT) > $(SYMS2)
103 $(GREP) -v UNDEF | $(AWK) '{print $$9}' | $(SORT) > $(SYMS2)
104 $(ECHO) "# GENERATED FILE - DO NOT EDIT" > @$
105 $(GREP) MAP-HEAD $(TEMPLATE2) | \
106 $(SED) -e "s/ *# MAP-HEAD//" >> @$
107 $(DIFF) $(SYMS1) $(SYMS2) | $(GREP) "^<" | \
108 $(GREP) -e "s/^< \(.*\)/ \1;/" >> @$
109 $(GREP) MAP-TAIL $(TEMPLATE2) | \
110 $(SED) -e "s/ *# MAP-TAIL//" >> @$
111 $(RM) $(SYMS1) $(SYMS2) $(MAIN1) $(MAIN2)

```

```

*****
11031 Sun May 26 18:22:59 2013
new/usr/src/head/Makefile
XXX Remove awk(1)
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2010 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # head/Makefile
26 #
27 # include global definitions
28 include ../Makefile.master

30 sparc_HDRS=
31 i386_HDRS=    stack_unwind.h

33 # Headers are listed one per line so that TeamWare can auto-merge most changes

35 KRB5HDRS= mit_copyright.h mit-sipb-copyright.h

37 ATTRDB_HDRS= secdb.h auth_attr.h exec_attr.h prof_attr.h user_attr.h \
38    auth_list.h

40 HDRS=  $(($(MACH)_HDRS) $(ATTRDB_HDRS) \
41    aio.h \
42    alloca.h \
43    apptrace.h \
44    apptrace_impl.h \
45    ar.h \
46    archives.h \
47    assert.h \
48    atomic.h \
49    attr.h \
50    config_admin.h \
51    cpio.h \
52    crypt.h \
53    ctype.h \
54    default.h \
55    devid.h \
56    devmgmt.h \
57    devpoll.h \
58    dial.h \
59    dirent.h \
60    dlfcn.h \
61    door.h \

```

```

62    elf.h \
63    err.h \
64    errno.h \
65    euc.h \
66    exacct.h \
67    exacct_impl.h \
68    execinfo.h \
69    fatal.h \
70    fcntl.h \
71    float.h \
72    fmtmsg.h \
73    fnmatch.h \
74    ftw.h \
75    gelf.h \
76    getopt.h \
77    getwidth.h \
78    glob.h \
79    grp.h \
80    iconv.h \
81    ieeeep.h \
82    ifaddrs.h \
83    inttypes.h \
84    iso646.h \
85    klpd.h \
86    langinfo.h \
87    lastlog.h \
88    lber.h \
89    ldap.h \
90    libelf.h \
91    libgen.h \
92    libintl.h \
93    libw.h \
94    libzonecfg.h \
95    limits.h \
96    linenum.h \
97    link.h \
98    listen.h \
99    locale.h \
100    macros.h \
101    malloc.h \
102    mdmn_changelog.h \
103    memory.h \
104    meta.h \
105    meta_runtime.h \
106    metadyn.h \
107    mon.h \
108    monetary.h \
109    mp.h \
110    mqueue.h \
111    nan.h \
112    ndbm.h \
113    ndpd.h \
114    netconfig.h \
115    netdb.h \
116    netdir.h \
117    nl_types.h \
118    nlist.h \
119    note.h \
120    nsctl.h \
121    nsswitch.h \
122    nss_common.h \
123    nss_dbdefs.h \
124    nss_netdir.h \
125    paths.h \
126    pcsample.h \
127    pfmt.h \

```

new/usr/src/head/Makefile

```

128     pkgdev.h      \
129     pkginfo.h     \
130     pkglocs.h     \
131     pkgstrct.h    \
132     pkgtrans.h    \
133     poll.h        \
134     port.h        \
135     priv.h        \
136     priv_utils.h  \
137     proc_service.h \
138     procfs.h      \
139     prof.h        \
140     project.h     \
141     pthread.h     \
142     pw.h          \
143     pwd.h         \
144     rctl.h        \
145     re_comp.h     \
146     regex.h       \
147     regexp.h      \
148     resolv.h      \
149     rje.h         \
150     rtdb_db.h     \
151     sac.h         \
152     sched.h       \
153     schedctl.h   \
154     sdssc.h       \
155     search.h      \
156     semaphore.h  \
157     setjmp.h      \
158     sgtty.h       \
159     shadow.h      \
160     siginfo.h     \
161     signal.h      \
162     spawn.h       \
163     stdarg.h      \
164     stdbool.h     \
165     stddef.h      \
166     stdint.h      \
167     stdio.h       \
168     stdio_ext.h   \
169     stdio_tag.h   \
170     stdio_impl.h  \
171     stdlib.h      \
172     storclass.h   \
173     string.h      \
174     strings.h     \
175     stropts.h     \
176     synch.h       \
177     sysexits.h    \
178     syslog.h      \
179     syms.h        \
180     tar.h         \
181     termio.h      \
182     termios.h     \
183     thread.h      \
184     thread_db.h  \
185     time.h        \
186     tiuser.h     \
187     tzfile.h      \
188     ucontext.h    \
189     ucred.h       \
190     ulimit.h      \
191     unistd.h      \
192     userdefs.h    \
193     ustat.h       \

```

3

new/usr/src/head/Makefile

```

194     utime.h       \
195     utmp.h        \
196     utmpx.h       \
197     valtools.h    \
198     values.h      \
199     varargs.h     \
200     wait.h        \
201     wchar.h       \
202     wchar_impl.h  \
203     wctype.h      \
204     widec.h       \
205     wordexp.h     \
206     xti.h         \
207     xti_inet.h    \
208     zone.h        \
209
210 ISOHDRS = \
211     ctype_c99.h   \
212     ctype_iso.h   \
213     limits_iso.h  \
214     locale_iso.h  \
215     setjmp_iso.h  \
216     signal_iso.h  \
217     stdarg_c99.h  \
218     stdarg_iso.h  \
219     stddef_iso.h  \
220     stdio_c99.h   \
221     stdio_iso.h   \
222     stdlib_c99.h  \
223     stdlib_iso.h  \
224     string_iso.h  \
225     time_iso.h    \
226     wchar_c99.h   \
227     wchar_iso.h   \
228     wctype_c99.h  \
229     wctype_iso.h  \
230
231 ARPAHDRS = \
232     ftp.h         \
233     inet.h        \
234     nameser.h     \
235     telnet.h      \
236     tftp.h        \
237     nameser_compat.h \
238
239 AUDIOHDRS = \
240     au.h          \
241
242 UUIDHDRS = \
243     uuid.h        \
244
245 # rpcsvc headers which are just headers (not derived from a .x file)
246 RPCSVC_SRC_HDRS = \
247     bootparam.h   \
248     daemon_utils.h \
249     dbm.h          \
250     nis_db.h       \
251     nislib.h       \
252     svc_dg_priv.h  \
253     yp_prot.h      \
254     ypclnt.h       \
255     yppasswd.h     \
256     ypuuid.h       \
257     rpc_sotypes.h  \
258
259 # rpcsvc headers which are generated from .x files

```

4

new/usr/src/head/Makefile

```

260 RPCSVC_GEN_HDRS = \
261     bootparam_prot.h \
262     mount.h \
263     nfs_prot.h \
264     nfs4_prot.h \
265     nis.h \
266     nlm_prot.h \
267     rex.h \
268     rquota.h \
269     rstat.h \
270     rusers.h \
271     rwall.h \
272     spray.h \
273     ufs_prot.h \
274     nfs_acl.h

276 LVMRPCHDRS = \
277     mhd.x mdi.o.x meta_basic.h metad.h metamed.h metamhd.h metacl.h

279 SYMHDRASSERT = $(ROOT)/usr/include/iso/assert_iso.h
280 SYMHDRERRNO = $(ROOT)/usr/include/iso/errno_iso.h
281 SYMHDRFLOAT = $(ROOT)/usr/include/iso/float_iso.h
282 SYMHDRISO646 = $(ROOT)/usr/include/iso/iso646_iso.h

284 RPCGENFLAGS = -C -h
285 rpcsvc/rwall.h := RPCGENFLAGS += -M
286 meta_basic.h := RPCGENFLAGS += -M
287 metad.h := RPCGENFLAGS += -M
288 metamed.h := RPCGENFLAGS += -M
289 mhd.x.h := RPCGENFLAGS += -M
290 mdi.o.x.h := RPCGENFLAGS += -M
291 metamhd.h := RPCGENFLAGS += -M
292 metacl.h := RPCGENFLAGS += -M

294 # rpcsvc rpcgen source (.x files)
295 #
296 # yp.x is an attempt at codifying what was hand coded in RPCL.
297 # Unfortunately it doesn't quite work. (The handcoded stuff isn't
298 # expressible in RPCL) this is due to the fact that YP was written
299 # before rpcgen existed. Hence, yp_prot.h cannot be derived from yp.x
300 #
301 # There is no '.h' for nis_object.x because it is included by nis.x and
302 # the resulting .h is nis.h.

304 RPCSVC_PROTS = \
305     $(RPCSVC_GEN_HDRS:%.h=%.x)      nis_object.x      yp.x

307 LVMSVC_PROTS = \
308     $(LVMRPCHDRS:%.h=%.x)

310 RPCSVCHDRS= $(RPCSVC_SRC_HDRS) $(RPCSVC_GEN_HDRS)

312 PROTOHDRS=  dumprestore.h routed.h ripngd.h rwhod.h timed.h

314 ROOTHDRS= $(HDRS:%=$(ROOT)/usr/include/%) \
315     $(KRB5HDRS:%=$(ROOT)/usr/include/kerberos5/%) \
316     $(ISOHDRS:%=$(ROOT)/usr/include/iso/%) \
317     $(ARPAHDRS:%=$(ROOT)/usr/include/arpa/%) \
318     $(AUDIOHDRS:%=$(ROOT)/usr/include/audio/%) \
319     $(UIDHDRS:%=$(ROOT)/usr/include/uid/%) \
320     $(RPCSVCHDRS:%=$(ROOT)/usr/include/rpcsvc/%) \
321     $(RPCSVC_PROTS:%=$(ROOT)/usr/include/rpcsvc/%) \
322     $(LVMRPCHDRS:%=$(ROOT)/usr/include/%) \
323     $(PROTOHDRS:%=$(ROOT)/usr/include/protocols/%)

325 DIRS= iso arpa audio rpcsvc protocols security uuid kerberosv5

```

5

new/usr/src/head/Makefile

```

326 ROOTDIRS= $(DIRS:%=$(ROOT)/usr/include/%)

328 SED= sed

330 # check files really don't exist
331 #
332 # should do something with the rpcsvc headers

334 iso/%.check:      iso/%.h
335     $(DOT_H_CHECK)

337 arpa/%.check:     arpa/%.h
338     $(DOT_H_CHECK)

340 audio/%.check:    audio/%.h
341     $(DOT_H_CHECK)

343 rpcsvc/%.check:   rpcsvc/%.h
344     $(DOT_H_CHECK)

346 rpcsvc/%.check:   rpcsvc/%.x
347     $(DOT_X_CHECK)

349 protocols/%.check: protocols/%.h
350     $(DOT_H_CHECK)

352 kerberosv5/%.check: kerberosv5/%.h
353     $(DOT_H_CHECK)

355 uuid/%.check:     uuid/%.h
356     $(DOT_H_CHECK)

358 # Note that the derived headers (rpcgen) are not checked at this time. These
359 # need work at the source level and rpcgen itself has a bug which causes a
360 # cstyle violation. Furthermore, there seems to be good reasons for the
361 # generated headers to not pass all of the hdrchk rules.
362 #
363 # Add the following to the CHECKHDRS list to activate the .x checks:
364 #     $(RPCSVC_PROTS:%.x=rpcsvc/%.check) \
365 #
366 CHECKHDRS= $(HDRS:%.h=%.check) \
367     $(KRB5HDRS:%.h=kerberosv5/%.check) \
368     $(ISOHDRS:%.h=iso/%.check) \
369     $(ARPAHDRS:%.h=arpa/%.check) \
370     $(AUDIOHDRS:%.h=audio/%.check) \
371     $(UIDHDRS:%.h=uid/%.check) \
372     $(RPCSVC_SRC_HDRS:%.h=rpcsvc/%.check) \
373     $(PROTOHDRS:%.h=protocols/%.check)

375 # headers which won't quite meet the standards...
376 #
377 # assert.h is required by ansi-c to *not* be idempotent (section 4.1.2).
378 # Hence the trailing guard is not the last thing in the file nor can it
379 # be without playing silly games.

381 assert.check      := HDRCHK_TAIL = | grep -v "end guard wrong" | true

383 # install rules

385 $(ROOT)/usr/include/security/%: security/%
386     $(INS.file)

388 $(ROOT)/usr/include/protocols/%: protocols/%
389     $(INS.file)

391 $(ROOT)/usr/include/rpcsvc/%: rpcsvc/%

```

6

new/usr/src/head/Makefile

7

```

392     $(INS.file)
394 $(ROOT)/usr/include/kerberos5/%: kerberos5/%
395     $(INS.file)
397 $(ROOT)/usr/include/arpa/%: arpa/%
398     $(INS.file)
400 $(ROOT)/usr/include/audio/%: audio/%
401     $(INS.file)
403 $(ROOT)/usr/include/iso/%: iso/%
404     $(INS.file)
406 $(ROOT)/usr/include/uuid/%: uuid/%
407     $(INS.file)
409 $(ROOT)/usr/include/%: %
410     $(INS.file)
412 .KEEP_STATE:
414 .PARALLEL:    $(ROOTHDRS) $(CHECKHDRS)
416 install_h:    $(ROOTDIRS) .WAIT $(ROOTHDRS) $(SYMHDRASSERT) $(SYMHDRERRNO) \
417     $(SYMHDRFLOAT) $(SYMHDRISO646)
419 check:    $(CHECKHDRS)
421 clean clobber:
422     $(RM) $(LVMRPCHDRS);
423     cd rpcsvc ; $(RM) $(RPCSVC_GEN_HDRS)
425 $(ROOTDIRS):
426     $(INS.dir)
428 $(SYMHDRASSERT):
429     -$(RM) $@; $(SYMLINK) ../assert.h $@
431 $(SYMHDRERRNO):
432     -$(RM) $@; $(SYMLINK) ../errno.h $@
434 $(SYMHDRFLOAT):
435     -$(RM) $@; $(SYMLINK) ../float.h $@
437 $(SYMHDRISO646):
438     -$(RM) $@; $(SYMLINK) ../iso646.h $@
440 rpcsvc/%.h:    rpcsvc/%.x
441     $(RPCGEN) $(RPCGENFLAGS) $< -o $@
443 rpcsvc/nis.h:  rpcsvc/nis.x
444     $(RPCGEN) $(RPCGENFLAGS) rpcsvc/nis.x | \
445     $(SED) -e '/EDIT_START/,,$$ d' > $@
447 meta_basic.h:  ../uts/common/sys/lvm/meta_basic.x
448     $(RPCGEN) $(RPCGENFLAGS) ../uts/common/sys/lvm/meta_basic.x | \
449     awk '/<synch.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t"
450     /<thread.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t/* _
451     { print $0 } \
452     ' > $@
454 metad.h:      metad.x
455     $(RPCGEN) $(RPCGENFLAGS) metad.x | \
456     awk '/<synch.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t"
457     /<thread.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t/* _

```

new/usr/src/head/Makefile

8

```

458     { print $0 } \
459     ' > $@
461 mhd_x.h:      ../uts/common/sys/lvm/mhd_x.x
462     $(RPCGEN) $(RPCGENFLAGS) ../uts/common/sys/lvm/mhd_x.x | \
463     awk '/<synch.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t"
464     /<thread.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t/* _
465     { print $0 } \
466     ' > $@
468 mdiox.h:     ../uts/common/sys/lvm/mdiox.x
469     $(RPCGEN) $(RPCGENFLAGS) ../uts/common/sys/lvm/mdiox.x | \
470     /usr/xpg4/bin/awk '{sub(/sys\/lvm\/md_mhd_x/, "mhd_x"); print $$0}' | \
471     /usr/xpg4/bin/awk '{sub(/sys\/lvm\/md_basic/, "meta_basic"); print $$0}'
472     nawk '{sub(/sys\/lvm\/md_mhd_x/, "mhd_x"); print $$0}' | \
473     nawk '{sub(/sys\/lvm\/md_basic/, "meta_basic"); print $$0}' | \
474     awk '/<synch.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t"
475     /<thread.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t/* _
476     { print $0 } \
477     ' > $@
477 metamed.h:   ../uts/common/sys/lvm/metamed.x
478     $(RPCGEN) $(RPCGENFLAGS) ../uts/common/sys/lvm/metamed.x | \
479     /usr/xpg4/bin/awk '{sub(/sys\/lvm\/md_basic/, "meta_basic"); print $$0}'
480     nawk '{sub(/sys\/lvm\/md_basic/, "meta_basic"); print $$0}' | \
481     awk '/<synch.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t"
482     /<thread.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t/* _
483     { print $0 } \
484     ' > $@
485 metamhd.h:   metamhd.x
486     $(RPCGEN) $(RPCGENFLAGS) metamhd.x | \
487     awk '/<synch.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t"
488     /<thread.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t/* _
489     { print $0 } \
490     ' > $@
492 metacl.h:    metacl.x
493     $(RPCGEN) $(RPCGENFLAGS) metacl.x | \
494     awk '/<synch.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t"
495     /<thread.h>/ { print "#ifdef _REENTRANT"; print $$0; print "#endif\t/* _
496     { print $0 } \
497     ' > $@

```

new/usr/src/lib/Makefile.lib

1

```
*****
8881 Sun May 26 18:22:59 2013
new/usr/src/lib/Makefile.lib
XXX Remove awk(1)
*****
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) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
22 #
23 #
24 # Definitions common to libraries.
25 #
26 # include global definitions; SRC should be defined in the shell.
27 # SRC is needed until RFE 1026993 is implemented.

29 include      $(SRC)/Makefile.master

31 ORDER=       lorder
32 TSORT=       tsort
33 AWK=         /usr/xpg4/bin/awk
34 AWK=         awk

35 #
36 # By default, we define the source directory for libraries to be
37 # one level up from the ISA-specific directory, where the code is
38 # actually built. Many libraries define a 'common' directory to
39 # contain the source. These libraries must redefine SRCDIR as:
40 #   SRCDIR = ../common
41 # Other variations are possible (../port, ../src, etc).
42 #
43 SRCDIR =     ..

45 #
46 # We define MAPFILES here for the benefit of most libraries, those that
47 # follow the convention of having source files and other common files
48 # in the $(SRCDIR) directory. Libraries that do not follow this
49 # convention must define MAPFILES, or MAPFILEDIR for themselves.
50 # Libraries that do follow this convention but that need supplemental
51 # ISA-specific mapfiles can augment MAPFILES like this:
52 #   MAPFILES += mapfile-vers
53 #
54 MAPFILEDIR = $(SRCDIR)
55 MAPFILES =   $(MAPFILEDIR)/mapfile-vers

57 #
58 # If HDRDIR is left unset, then it's possible for the $(ROOTHDRDIR)/%
59 # install rule in lib/Makefile.targ to generate false matches if there
60 # are any common directory names between / and /usr/include ('xfn' is
```

new/usr/src/lib/Makefile.lib

2

```
61 # one common example). To prevent this, we set HDRDIR to a directory
62 # name that will almost surely not exist on the build machine.
63 #
64 HDRDIR=      /__nonexistent_directory__

66 #
67 # We don't build archive (*.a) libraries by default anymore.
68 # If a component of the build needs to build an archive library
69 # for its own internal purposes, it can define LIBS for itself
70 # after including Makefile.lib, like this:
71 #   LIBS = $(LIBRARY)
72 # or:
73 #   LIBS = $(LIBRARYCCC)
74 # Archive libraries must not be installed in the proto area.
75 #
76 LIBS=
77 MACHLIBS=    $(LIBS:%=$(MACH)/%)
78 MACHLIBS64=  $(LIBS:%=$(MACH64)/%)
79 DYNLIB=      $(LIBRARY:.a=.so$(VERS))
80 DYNLIBPSR=   $(LIBRARY:.a=_psr.so$(VERS))
81 DYNLIBCCC=   $(LIBRARYCCC:.a=.so$(VERS))
82 LIBLINKS=    $(LIBRARY:.a=.so)
83 LIBLINKSCCC= $(LIBRARYCCC:.a=.so)
84 LIBNAME=     $(LIBRARY:lib%.a=%)
85 LIBLINKPATH=
86 LIBNULL=    null.a
87 ROOTHDRDIR=  $(ROOT)/usr/include
88 ROOTLIBDIR=  $(ROOT)/usr/lib
89 ROOTLIBDIR64= $(ROOT)/usr/lib/$(MACH64)
90 ROOTFS_LIBDIR= $(ROOT)/lib
91 ROOTFS_LIBDIR64= $(ROOT)/lib/$(MACH64)
92 ROOTLINTDIR= $(ROOTLIBDIR)
93 ROOTFS_LINTDIR= $(ROOTFS_LIBDIR)
94 ROOTFS_LINTDIR64= $(ROOTFS_LIBDIR64)
95 ROOTHDRS=    $(HDRS:%=$(ROOTHDRDIR)/%)
96 HDRSRCS=     $(HDRS:%=$(HDRDIR)/%)
97 CHECKHDRS=   $(HDRSRCS:%.h=%_check)
98 ROOTLIBS=    $(LIBS:%=$(ROOTLIBDIR)/%)
99 ROOTLIBS64=  $(LIBS:%=$(ROOTLIBDIR64)/%)
100 ROOTFS_LIBS= $(DYNLIB:%=$(ROOTFS_LIBDIR)/%)
101 ROOTFS_LIBS64= $(DYNLIB:%=$(ROOTFS_LIBDIR64)/%)
102 ROOTLINKS=   $(ROOTLIBDIR)/$(LIBLINKS)
103 ROOTLINKS64= $(ROOTLIBDIR64)/$(LIBLINKS)
104 ROOTFS_LINKS= $(ROOTFS_LIBDIR)/$(LIBLINKS)
105 ROOTFS_LINKS64= $(ROOTFS_LIBDIR64)/$(LIBLINKS)
106 ROOTLINKSCCC= $(ROOTLIBDIR)/$(LIBLINKSCCC)
107 ROOTLINKSCCC64= $(ROOTLIBDIR64)/$(LIBLINKSCCC)
108 ROOTFS_LINKSCCC= $(ROOTFS_LIBDIR)/$(LIBLINKSCCC)
109 ROOTFS_LINKSCCC64= $(ROOTFS_LIBDIR64)/$(LIBLINKSCCC)
110 ROOTLINT=    $(LINTSRC:%=$(ROOTLINTDIR)/%)
111 ROOTFS_LINT= $(LINTSRC:%=$(ROOTFS_LINTDIR)/%)
112 ROOTFS_LINT64= $(LINTSRC:%=$(ROOTFS_LINTDIR64)/%)
113 ROOTMAN3=    $(ROOT)/usr/share/man/man3
114 ROOTMAN3FILES= $(MAN3FILES:%=$(ROOTMAN3)/%)
115 $(ROOTMAN3FILES) := FILEMODE= 444

117 # Demo rules
118 DEMOFILES=
119 DEMOFILESRCDIR= common
120 ROOTDEMODIRBASE= __nonexistent_directory__
121 ROOTDEMODIRS=
122 ROOTDEMFILES= $(DEMOFILES:%=$(ROOTDEMODIRBASE)/%)
123 $(ROOTDEMODIRS) := DIRMODE = 755

125 LINTLIB=     llib-1$(LIBNAME).ln
126 LINTFLAGS=   -uaxm
```

new/usr/src/lib/Makefile.lib

```

127 LINTFLAGS64=      -uaxm -m64
128 LINTSRC=          $(LINTLIB:%.ln=%)
129 LINTOUT=          lint.out
130 ARFLAGS=          r
131 SONAME=            $(DYNLIB)
132 # For most libraries, we should be able to resolve all symbols at link time,
133 # either within the library or as dependencies, all text should be pure, and
134 # combining relocations into one relocation table reduces startup costs.
135 # All options are tunable to allow overload/omission from lower makefiles.

138 HSONAME=          -h$(SONAME)
139 DYNFLAGS=          $(HSONAME) $(ZTEXT) $(ZDEFS) $(BDIRECT) \
140                  $(MAPFILES:%=-M%) $(MAPFILE.PGA:%=-M%) $(MAPFILE.NED:%=-M%)

142 LDLIBS=           $(LDLIBS.lib)

144 OBJS=              $(OBJECTS:%=objs/%)
145 PICS=              $(OBJECTS:%=pics/%)

147 # Declare that all library .o's can all be made in parallel.
148 # The DUMMY target is for those instances where OBJS and PICS
149 # are empty (to avoid an unconditional .PARALLEL declaration).
150 .PARALLEL:         $(OBJS) $(PICS) DUMMY

152 # default value for "portable" source
153 SRCS=              $(OBJECTS:%.o=$(SRCDIR)/%.c)

155 # default build of an archive and a shared object,
156 # overridden locally when extra processing is needed
157 BUILD.AR=          $(AR) $(ARFLAGS) $@ $(AROBJ)
158 BUILD.SO=          $(CC) -o $@ $(GSHARED) $(DYNFLAGS) $(PICS) $(EXTPICS) $(LDLIBS)
159 BUILD.DCC.SO=      $(CCC) -o $@ $(GSHARED) $(DYNFLAGS) $(PICS) $(EXTPICS) $(LDLIBS)

161 # default dynamic library symlink
162 # IMPORTANT:: If you change INS.liblink OR INS.liblink64 here, then you
163 # MUST also change the corresponding override definitions in
164 # $CLOSED/Makefile.tonic.
165 #
166 # If you do not do this, then the closedbins build for the OpenSolaris
167 # community will break. PS, the gatekeepers will be upset too.
168 #
169 INS.liblink=        -$(RM) $@; $(SYMLINK) $(LIBLINKPATH)$ (LIBLINKS)$ (VERS) $@
170 INS.liblinkccc=     -$(RM) $@; $(SYMLINK) $(LIBLINKPATH)$ (LIBLINKSCCC)$ (VERS) $@

172 # default 64-bit dynamic library symlink
173 INS.liblink64=      -$(RM) $@; $(SYMLINK) $(LIBLINKPATH)$ (LIBLINKS)$ (VERS) $@
174 INS.liblinkccc64=   -$(RM) $@; $(SYMLINK) $(LIBLINKPATH)$ (LIBLINKSCCC)$ (VERS) $@

176 #
177 # If appropriate, augment POST_PROCESS_O and POST_PROCESS_SO to do CTF
178 # processing. We'd like to just conditionally append to POST_PROCESS_O and
179 # POST_PROCESS_SO, but ParallelMake has a bug which causes the same value to
180 # sometimes get appended more than once, which will cause ctffconvert to fail.
181 # So, instead we introduce CTFCONVERT_POST and CTFMERGE_POST, which are always
182 # appended to POST_PROCESS_O and POST_PROCESS_SO but are no-ops unless CTF
183 # processing should be done.
184 #
185 CTFCONVERT_POST = :
186 CTFMERGE_POST    = :
187 POST_PROCESS_O += ; $(CTFCONVERT_POST)
188 POST_PROCESS_SO += ; $(CTFMERGE_POST)

190 CTFMERGE_LIB      = $(CTFMERGE) -t -f -L VERSION -o $@ $(PICS)

192 # conditional assignments

```

3

new/usr/src/lib/Makefile.lib

```

194 $(OBJ) :=          sparc_CFLAGS += -xregs=no%appl

196 $(PICS) :=         sparc_CFLAGS += -xregs=no%appl $(sparc_C_PICFLAGS)
197 $(PICS) :=         sparcv9_CFLAGS += -xregs=no%appl $(sparcv9_C_PICFLAGS)
198 $(PICS) :=         i386_CFLAGS += $(i386_C_PICFLAGS)
199 $(PICS) :=         amd64_CFLAGS += $(amd64_C_PICFLAGS)
200 $(PICS) :=         CCFLAGS += $(CC_PICFLAGS)
201 $(PICS) :=         CPPFLAGS += -DPIC -D_REENTRANT
202 $(PICS) :=         sparcv9_CCFLAGS += -xregs=no%appl $(sparcv9_CC_PICFLAGS)
203 $(PICS) :=         amd64_CCFLAGS += $(amd64_CC_PICFLAGS)
204 $(PICS) :=         CFLAGS += $(CTF_FLAGS)
205 $(PICS) :=         CFLAGS64 += $(CTF_FLAGS)
206 $(PICS) :=         CTFCONVERT_POST = $(CTFCONVERT_O)
207 $(DYNLIB) :=       CTFMERGE_POST = $(CTFMERGE_LIB)

209 $(LINTLIB):=       LOG = -DLOGGING
210 $(LIBRARY):=       AROBJ = $(OBJ)
211 $(LIBRARY):=       DIR = objs
212 $(DYNLIB):=        DIR = pics
213 $(DYNLIBBCC):=     DIR = pics

215 SONAMECCC=         $(DYNLIBBCC)
216 HSONAMECCC=         -h $(SONAMECCC)
217 #
218 # Keep in sync with the standard DYNFLAGS
219 #
220 $(DYNLIBBCC):=      DYNFLAGS = $(HSONAMECCC) $(ZTEXT) $(ZDEFS) \
221                      $(MAPFILES:%=-M%) $(MAPFILE.PGA:%=-M%) $(MAPFILE.NED:%=-M%) \
222                      $(BDIRECT) $(NORUNPATH)

225 # build rule for "portable" source
226 objs/%.o pics/%.o: %.c
227     $(COMPILE.c) -o $@ $<
228     $(POST_PROCESS_O)

230 objs/%.o pics/%.o: %.cc
231     $(COMPILE.cc) -o $@ $<
232     $(POST_PROCESS_O)

234 .PRECIOUS: $(LIBS)

236 # Define the majority text domain in this directory.
237 TEXT_DOMAIN= SUNW_OST_OSLIB

239 $(ROOTMAN3)/%: % .sunman
240     $(INS.rename)

242 #
243 # For library source code, we expect that some symbols may not be used or
244 # may *appear* to be able to rescope to static; shut lint up. Never add
245 # a flag here unless you're *sure* that all libraries need to be linted
246 # with it.
247 #
248 LINTCHECKFLAGS = -m -erroff=E_NAME_DEF_NOT_USED2
249 LINTCHECKFLAGS += -erroff=E_NAME_DECL_NOT_USED_DEF2

251 #
252 # Target Architecture
253 #
254 TARGETMACH=        $(MACH)

256 #
257 # Allow people to define their own clobber rules. Normal makefiles
258 # shouldn't override this - they should override $(CLOBBERFILES) instead.

```

4

new/usr/src/lib/Makefile.lib

5

```
259 #  
260 CLOBBERTARGETFILES= $(LIBS) $(DYNLIB) $(CLOBBERFILES)
```

```

*****
26363 Sun May 26 18:23:11 2013
new/usr/src/lib/brand/shared/zone/common.ksh
XXX Remove awk(1)
*****
_____unchanged_portion_omitted_____

237 #
238 # Read zonecfg fs entries and save the relevant data, one entry per
239 # line.
240 # This assumes the properties from the zonecfg output, e.g.:
241 #     fs:
242 #         dir: /opt
243 #         special: /opt
244 #         raw not specified
245 #         type: lofs
246 #         options: [noexec,ro,noatime]
247 #
248 # and it assumes the order of the fs properties as above.
249 #
250 get_fs_info()
251 {
252     zonecfg -z $zonename info fs | /usr/xpg4/bin/awk '{
252     zonecfg -z $zonename info fs | awk '{
253     if ($1 == "options:") {
254         # Remove brackets.
255         options=substr($2, 2, length($2) - 2);
256         printf("%s %s %s %s\n", dir, type, special, options);
257     } else if ($1 == "dir:") {
258         dir=$2;
259     } else if ($1 == "special:") {
260         special=$2;
261     } else if ($1 == "type:") {
262         type=$2
263     }
264     }' >> $fstmpfile
265 }

267 #
268 # Mount zonecfg fs entries into the zonepath.
269 #
270 mnt_fs()
271 {
272     if [ ! -s $fstmpfile ]; then
273         return;
274     fi

276     # Sort the fs entries so we can handle nested mounts.
277     sort $fstmpfile | /usr/xpg4/bin/awk -v zonepath=$zonepath '{
277     sort $fstmpfile | awk -v zonepath=$zonepath '{
278         if (NF == 4)
279             options="-o " $4;
280         else
281             options=""

283         # Create the mount point. Ignore errors since we might have
284         # a nested mount with a pre-existing mount point.
285         cmd="/usr/bin/mkdir -p " zonepath "/"root" $1 " >/dev/null 2>&1"
286         system(cmd);

288         cmd="/usr/sbin/mount -F " $2 " " options " " $3 " " \
289             zonepath "/"root" $1;
290         if (system(cmd) != 0) {
291             printf("command failed: %s\n", cmd);
292             exit 1;
293         }

```

```

294     }' >>$LOGFILE
295 }

297 #
298 # Unmount zonecfg fs entries from the zonepath.
299 #
300 umnt_fs()
301 {
302     if [ ! -s $fstmpfile ]; then
303         return;
304     fi

306     # Reverse sort the fs entries so we can handle nested unmounts.
307     sort -r $fstmpfile | /usr/xpg4/bin/awk -v zonepath=$zonepath '{
307     sort -r $fstmpfile | awk -v zonepath=$zonepath '{
308         cmd="/usr/sbin/umount " zonepath "/"root" $1
309         if (system(cmd) != 0) {
310             printf("command failed: %s\n", cmd);
311         }
312     }' >>$LOGFILE
313 }

315 # Find the dataset mounted on the zonepath.
316 get_zonepath_ds() {
317     ZONEPATH_DS=\usr/sbin/zfs list -H -t filesystem -o name,mountpoint | \
318     /usr/xpg4/bin/awk -v zonepath=$1 '{
318     /usr/bin/awk -v zonepath=$1 '{
319         if ($2 == zonepath)
320             print $1
321     }'

323     if [ -z "$ZONEPATH_DS" ]; then
324         fail_fatal "$f_no_ds"
325     fi
326 }

_____unchanged_portion_omitted_____

528 #
529 # Get the archive base.
530 #
531 # We must unpack the archive in the right place within the zonepath so
532 # that files are installed into the various mounted filesystems that are set
533 # up in the zone's configuration. These are already mounted for us by the
534 # mntfs function.
535 #
536 # Archives can be made of either a physical host's root file system or a
537 # zone's zonepath. For a physical system, if the archive is made using an
538 # absolute path (/..) we can't use it. For a zone the admin can make the
539 # archive from a variety of locations;
540 #
541 # a) zonepath itself: This will be a single dir, probably named with the
542 # zone name, it will contain a root dir and under the root we'll see all
543 # the top level dirs; etc, var, usr... We must be above the ZONEPATH
544 # when we unpack the archive but this will only work if the the archive's
545 # top-level dir name matches the ZONEPATH base-level dir name. If not,
546 # this is an error.
547 #
548 # b) inside the zonepath: We'll see root and it will contain all the top
549 # level dirs; etc, var, usr.... We must be in the ZONEPATH when we unpack
550 # the archive.
551 #
552 # c) inside the zonepath root: We'll see all the top level dirs, ./etc,
553 # ./var, ./usr.... This is also the case we see when we get an archive
554 # of a physical system. We must be in ZONEROOT when we unpack the archive.
555 #
556 # Note that there can be a directory named "root" under the ZONEPATH/root

```

```

557 # directory.
558 #
559 # This function handles the above possibilities so that we reject absolute
560 # path archives and figure out where in the file system we need to be to
561 # properly unpack the archive into the zone. It sets the ARCHIVE_BASE
562 # variable to the location where the archive should be unpacked.
563 #
564 get_archive_base()
565 {
566     stage1=$1
567     archive=$2
568     stage2=$3
569
570     vlog "$m_analyse_archive"
571
572     base=`$stage1 $archive | $stage2 2>/dev/null | /usr/xpg4/bin/awk -F/ '{
572     base=`$stage1 $archive | $stage2 2>/dev/null | nawk -F/ '{
573         # Check for an absolute path archive
574         if (substr($0, 1, 1) == "/")
575             exit 1
576
577         if ($1 != ".")
578             dirs[$1] = 1
579         else
580             dirs[$2] = 1
581     }
582     END {
583         for (d in dirs) {
584             cnt++
585             if (d == "bin") sawbin = 1
586             if (d == "etc") sawetc = 1
587             if (d == "root") sawroot = 1
588             if (d == "var") sawvar = 1
589         }
590
591         if (cnt == 1) {
592             # If only one top-level dir named root, we are in the
593             # zonepath, otherwise this must be an archive *of*
594             # the zonepath so print the top-level dir name.
595             if (sawroot)
596                 print "*zonepath*"
597             else
598                 for (d in dirs) print d
599         } else {
600             # We are either in the zonepath or in the zonepath/root
601             # (or at the top level of a full system archive which
602             # looks like the zonepath/root case). Figure out which
603             # one.
604             if (sawroot && !sawbin && !sawetc && !sawvar)
605                 print "*zonepath*"
606             else
607                 print "*zoneroot*"
608         }
609     },`
610
611     if (( $? != 0 )); then
612         umnt_fs
613         fatal "$e_absolute_archive"
614     fi
615
616     if [[ "$base" == "*zoneroot*" ]]; then
617         ARCHIVE_BASE=$ZONEROOT
618     elif [[ "$base" == "*zonepath*" ]]; then
619         ARCHIVE_BASE=$ZONEPATH
620     else
621         # We need to be in the dir above the ZONEPATH but we need to

```

```

622         # validate that $base matches the final component of ZONEPATH.
623         bname=`basename $ZONEPATH`
624
625         if [[ "$bname" != "$base" ]]; then
626             umnt_fs
627             fatal "$e_mismatch_archive" "$base" "$bname"
628         fi
629         ARCHIVE_BASE=`dirname $ZONEPATH`
630     fi
631 }

```

unchanged portion omitted

new/usr/src/lib/brand/shared/zone/uninstall.ksh

1

19427 Sun May 26 18:23:22 2013

new/usr/src/lib/brand/shared/zone/uninstall.ksh

XXX Remove awk(1)

unchanged_portion_omitted_

```
248 # find the dataset associated with $zonepath
249 uninstall_get_zonepath_ds()
250 {
251     ZONEPATH_DS=`/sbin/zfs list -t filesystem -o name,mountpoint | \
252     /usr/xpg4/bin/awk -v zonepath=$zonepath '{
253     /bin/nawk -v zonepath=$zonepath '{
253         if ($2 == zonepath)
254             print $1
255     }`
257     if [ -z "$ZONEPATH_DS" ]; then
258         # there is no $zonepath dataset
259         rm_zonepath
260         exit $ZONE_SUBPROC_OK
261     fi
262 }
```

unchanged_portion_omitted_

new/usr/src/lib/brand/solaris10/zone/common.ksh

1

```
*****
10768 Sun May 26 18:23:22 2013
new/usr/src/lib/brand/solaris10/zone/common.ksh
XXX Remove awk(1)
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2010 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #

26 unset LD_LIBRARY_PATH
27 PATH=/usr/bin:/usr/sbin
28 export PATH

30 . /usr/lib/brand/shared/common.ksh

32 # Values for service tags.
33 STCLIENT=/usr/bin/stclient
34 ST_PRODUCT_NAME="Solaris 10 Containers"
35 ST_PRODUCT_REV="1.0"
36 ST_PRODUCT_UUID="urn:uuid:2f459121-dec7-11de-9af7-080020a9ed93"

38 w_sanity_detail=$(gettext "      WARNING: Skipping image sanity checks.")
39 f_sanity_detail=$(gettext "Missing %s at %s")
40 f_sanity_sparse=$(gettext "Is this a sparse zone image? The image must be whol
41 f_sanity_vers=$(gettext "The image release version must be 10 (got %s), the zon
42 f_not_s10_image=$(gettext "%s doesn't look like a Solaris 10 image.")
43 f_sanity_nopatch=$(gettext "Unable to determine the image's patch level.")
44 f_sanity_downrev=$(gettext "The image patch level is downrev for running in a so
45 f_need_newer_emul=$(gettext "The image requires a newer version of the solaris10
46 f_zfs_create=$(gettext "Unable to create the zone's ZFS dataset.")
47 f_no_ds=$(gettext "No zonepath dataset; the zonepath must be a ZFS dataset.")
48 f_multiple_ds=$(gettext "Multiple active datasets.")
49 f_no_active_ds=$(gettext "No active dataset; the zone's ZFS root dataset must be
50 f_zfs_unmount=$(gettext "Unable to unmount the zone's root ZFS dataset (%s).\nIs
51 f_zfs_mount=$(gettext "Unable to mount the zone's ZFS dataset.")
52 incompat_options=$(gettext "mutually exclusive options.\n%s")

54 sanity_ok=$(gettext "      Sanity Check: Passed. Looks like a Solaris 10 image.
55 sanity_fail=$(gettext "      Sanity Check: FAILED (see log for details).")

57 e_badboot=$(gettext "Zone boot failed")
58 e_nosingleuser=$(gettext "ERROR: zone did not finish booting to single-user.")
59 e_unconfig=$(gettext "sys-unconfig failed")
60 v_unconfig=$(gettext "Performing zone sys-unconfig")
```

new/usr/src/lib/brand/solaris10/zone/common.ksh

2

```
62 v_no_tags=$(gettext "Service tags facility not present.")
63 e_bad_uuid=$(gettext "Failed to get zone UUID")
64 v_addtag=$(gettext "Adding service tag: %s")
65 v_deltag=$(gettext "Removing service tag: %s")
66 e_addtag_fail=$(gettext "Adding service tag failed (error: %s)")

68 sanity_check()
69 {
70     typeset dir="$1"
71     res=0

73     #
74     # Check for some required directories and make sure this isn't a
75     # sparse zone image.
76     #
77     checks="etc etc/svc var var/svc"
78     for x in $checks; do
79         if [[ ! -e $dir/$x ]]; then
80             log "$f_sanity_detail" "$x" "$dir"
81             res=1
82         fi
83     done
84     # Files from SUNWcsr and SUNWcsu that are in sparse inherit-pkg-dirs.
85     checks="lib/svc sbin/zonename usr/bin/chmod"
86     for x in $checks; do
87         if [[ ! -e $dir/$x ]]; then
88             log "$f_sanity_detail" "$x" "$dir"
89             log "$f_sanity_sparse"
90             res=1
91         fi
92     done

94     if (( $res != 0 )); then
95         log "$sanity_fail"
96         fatal "$install_fail" "$ZONENAME"
97     fi

99     if [[ "$SANITY_SKIP" == 1 ]]; then
100         log "$w_sanity_detail"
101         return
102     fi

104     #
105     # Check image release to be sure its S10.
106     #
107     image_vers="unknown"
108     if [[ -f $dir/var/sadm/system/admin/INST_RELEASE ]]; then
109         image_vers=$(/usr/xpg4/bin/awk -F= '{if ($1 == "VERSION") print
109         image_vers=$(nawk -F= '{if ($1 == "VERSION") print $2}' \
110             $dir/var/sadm/system/admin/INST_RELEASE)
111     fi

113     if [[ "$image_vers" != "10" ]]; then
114         log "$f_sanity_vers" "$image_vers"
115         res=1
116     fi

118     #
119     # Make sure we have the minimal KU patch we support. These are the
120     # KUs for S10u8.
121     #
122     if [[ $(uname -p) == "i386" ]]; then
123         req_patch="141445-09"
124     else
125         req_patch="141444-09"
126     fi
```

```

128     for i in $dir/var/sadm/pkg/SUNWcakr*
129     do
130         if [[ ! -d $i || ! -f $i/pkginfo ]]; then
131             log "$f_sanity_nopatch"
132             res=1
133         fi
134     done
135
136     #
137     # Check the core kernel pkg for the required KU patch.
138     #
139     found=0
140     for i in $dir/var/sadm/pkg/SUNWcakr*/pkginfo
141     do
142         patches=$( /usr/xpg4/bin/awk -F= '{if ($1 == "PATCHLIST") print $
142         patches=$(nawk -F= '{if ($1 == "PATCHLIST") print $2}' $i)
143         for patch in $patches
144         do
145             if [[ $patch == $req_patch ]]; then
146                 found=1
147                 break
148             fi
149         done
150
151         if (( $found == 1 )); then
152             break
153         fi
154     done
155
156     if (( $found != 1 )); then
157         log "$f_sanity_downrev" "$patches"
158         res=1
159     fi
160
161     #
162     # Check the S10 image for a required version of the emulation.
163     #
164     VERS_FILE=/usr/lib/brand/solaris10/version
165     s10vers_needs=0
166     if [[ -f $dir/$VERS_FILE ]]; then
167         s10vers_needs=$( /usr/bin/egrep -v "^#" $dir/$VERS_FILE)
168     fi
169
170     # Now get the current emulation version.
171     emul_vers=$( /usr/bin/egrep -v "^#" $VERS_FILE)
172
173     # Verify that the emulation can run this version of S10.
174     if (( $s10vers_needs > $emul_vers )); then
175         log "$f_need_newer_emul"
176         res=1
177     fi
178
179     if (( $res != 0 )); then
180         log "$sanity_fail"
181         fatal "$install_fail" "$ZONENAME"
182     fi
183
184     vlog "$sanity_ok"
185 }
186
187 unchanged_portion_omitted
188
189 309 #
190 310 # Get zone's uuid for service tag.
191 311 #
192 312 get_inst_uuid()

```

```

313 {
314     typeset ZONENAME="$1"
315
316     ZONEUUID='zoneadm -z $ZONENAME list -p | /usr/xpg4/bin/awk -F: '{print $
316     ZONEUUID='zoneadm -z $ZONENAME list -p | nawk -F: '{print $5}'
317     [[ $? -ne 0 || -z $ZONEUUID ]] && return 1
318
319     INSTANCE_UUID="urn:st:${ZONEUUID}"
320     return 0
321 }
322
323 unchanged_portion_omitted

```

new/usr/src/lib/brand/solaris10/zone/p2v.ksh

1

```
*****
17569 Sun May 26 18:23:22 2013
new/usr/src/lib/brand/solaris10/zone/p2v.ksh
XXX Remove awk(1)
*****
_____unchanged_portion_omitted_____

82 #
83 # For an exclusive stack zone, fix up the network configuration files.
84 # We need to do this even if unconfiguring the zone so sys-unconfig works
85 # correctly.
86 #
87 fix_net()
88 {
89     [[ "$STACK_TYPE" == "shared" ]] && return

91     NETIF_CNT=$( /usr/bin/ls $ZONEROOT/etc/hostname.* 2>/dev/null | \
92     /usr/bin/wc -l)
93     if (( $NETIF_CNT != 1 )); then
94         vlog "$v_nonetfix"
95         return
96     fi

98     NET=$(LC_ALL=C /usr/sbin/zonecfg -z $ZONENAME info net)
99     if (( $? != 0 )); then
100         error "$e_badinfo" "net"
101         return
102     fi

104     NETIF=$(echo $NET | /usr/xpg4/bin/awk '{
104     NETIF=$(echo $NET | /usr/bin/nawk '{
105         for (i = 1; i < NF; i++) {
106             if ($i == "physical:") {
107                 if (length(net) == 0) {
108                     i++
109                     net = $i
110                 } else {
111                     multiple=1
112                 }
113             }
114         }
115     }
116     END { if (!multiple)
117         print net
118     }')

120     if [[ -z "$NETIF" ]]; then
121         vlog "$v_nonetfix"
122         return
123     fi

125     OLD_HOSTNET=$( /usr/bin/ls $ZONEROOT/etc/hostname.*)
126     if [[ "$OLD_HOSTNET" != "$ZONEROOT/etc/hostname.$NETIF" ]]; then
127         safe_move $OLD_HOSTNET $ZONEROOT/etc/hostname.$NETIF
128     fi
129 }

131 #
132 # Disable all of the shares since the zone cannot be an NFS server.
133 # Note that we disable the various instances of the svc:/network/shares/group
134 # SMF service in the fix_smf function.
135 #
136 fix_nfs()
137 {
138     zonedfs=$ZONEROOT/etc/dfs
```

new/usr/src/lib/brand/solaris10/zone/p2v.ksh

2

```
140     [[ ! -d $zonedfs ]] && return

142     if [[ -h $zonedfs/dfstab || ! -f $zonedfs/dfstab ]]; then
143         error "$e_badfile" "/etc/dfs/dfstab"
144         return
145     fi

147     tmpfile=$(mktemp -t)
148     if [[ $? == 1 || -z "$tmpfile" ]]; then
149         error "$e_tmpfile"
150         return
151     fi

153     /usr/xpg4/bin/awk '{
153     /usr/bin/nawk '{
154         if (substr($1, 0, 1) == "#") {
155             print $0
156         } else {
157             print "#", $0
158             modified=1
159         }
160     }
161     END {
162         if (modified == 1) {
163             printf("# Modified by p2v ")
164             system("/usr/bin/date")
165             exit 0
166         }
167         exit 1
168     }' $zonedfs/dfstab >>$tmpfile

170     if (( $? == 0 )); then
171         if [[ ! -f $zonedfs/dfstab.pre_p2v ]]; then
172             safe_copy $zonedfs/dfstab $zonedfs/dfstab.pre_p2v
173         fi
174         safe_copy $tmpfile $zonedfs/dfstab
175         chown root:sys $zonedfs/dfstab || \
176             fail_fatal "$f_chown" "$zonedfs/dfstab"
177         chmod 644 $zonedfs/dfstab || \
178             fail_fatal "$f_chmod" "$zonedfs/dfstab"
179     fi
180     /usr/bin/rm -f $tmpfile
181 }

183 #
184 # Comment out most of the old mounts since they are either unneeded or
185 # likely incorrect within a zone. Specific mounts can be manually
186 # reenabled if the corresponding device is added to the zone.
187 #
188 fix_vfstab()
189 {
190     if [[ -h $ZONEROOT/etc/vfstab || ! -f $ZONEROOT/etc/vfstab ]]; then
191         error "$e_badfile" "/etc/vfstab"
192         return
193     fi

195     tmpfile=$(mktemp -t)
196     if [[ $? == 1 || -z "$tmpfile" ]]; then
197         error "$e_tmpfile"
198         return
199     fi

201     /usr/xpg4/bin/awk '{
201     /usr/bin/nawk '{
202         if (substr($1, 0, 1) == "#") {
203             print $0
```

```

204     } else if ($1 == "fd" || $1 == "/proc" || $1 == "swap" ||
205     $1 == "ctfs" || $1 == "objfs" || $1 == "sharefs" ||
206     $4 == "nfs" || $4 == "lofs") {
207         print $0
208     } else {
209         print "#", $0
210         modified=1
211     }
212 }
213 END {
214     if (modified == 1) {
215         printf("# Modified by p2v ")
216         system("/usr/bin/date")
217         exit 0
218     }
219     exit 1
220 }' $ZONEROOT/etc/vfstab >>$tmpfile

222 if (( $? == 0 )); then
223     if [[ ! -f $ZONEROOT/etc/vfstab.pre_p2v ]]; then
224         safe_copy $ZONEROOT/etc/vfstab \
225             $ZONEROOT/etc/vfstab.pre_p2v
226     fi
227     safe_copy $tmpfile $ZONEROOT/etc/vfstab
228     chown root:sys $ZONEROOT/etc/vfstab || \
229         fail_fatal "$f_chown" "$ZONEROOT/etc/vfstab"
230     chmod 644 $ZONEROOT/etc/vfstab || \
231         fail_fatal "$f_chmod" "$ZONEROOT/etc/vfstab"
232 fi
233 /usr/bin/rm -f $tmpfile
234 }

236 #
237 # Collect the data needed to delete SMF services. Since we're p2v-ing a
238 # physical image there are SMF services which must be deleted.
239 #
240 fix_smf_pre_uoa()
241 {
242     #
243     # Start by getting the svc manifests that are delivered by hollow
244     # pkgs then use 'svccfg inventory' to get the names of the svcs
245     # delivered by those manifests. The svc names are saved into a
246     # temporary file.
247     #

249     SMFTMPFILE=$(mktemp -t smf.XXXXXX)
250     if [[ $? == 1 || -z "$SMFTMPFILE" ]]; then
251         error "$e_tmpfile"
252         return
253     fi

255     for i in $ZONEROOT/var/sadm/pkg/*
256     do
257         pkg=$(/usr/bin/basename $i)
258         [[ ! -f $ZONEROOT/var/sadm/pkg/$pkg/save/pspool/$pkg/pkgmap ]] \
259             && continue

261         /usr/bin/egrep -s "SUNW_PKG_HOLLOW=true" \
262             $ZONEROOT/var/sadm/pkg/$pkg/pkginfo || continue

264         for j in $(/usr/xpg4/bin/awk '{if ($2 == "f" &&
264         for j in $(/usr/bin/nawk '{if ($2 == "f" &&
265         substr($4, 1, 17) == "var/svc/manifest/") print $4}' \
266         $ZONEROOT/var/sadm/pkg/$pkg/save/pspool/$pkg/pkgmap)
267     do
268         svcs=$(SVCCFG_NOVALIDATE=1 \

```

```

269         SVCCFG_REPOSITORY=$ZONEROOT/etc/svc/repository.db \
270         /usr/sbin/svccfg inventory $ZONEROOT/$j)
271         for k in $svcs
272         do
273             echo $k /$j >> $SMFTMPFILE
274         done
275     done
276 done
277 }

279 #
280 # Delete or disable SMF services.
281 # Zone is booted to milestone=none when this function is called.
282 # Use the SMF data collected by fix_smf_pre_uoa() to delete the services.
283 #
284 fix_smf()
285 {
286     #
287     # Zone was already booted to milestone=none, wait until SMF door exists.
288     #
289     for i in 0 1 2 3 4 5 6 7 8 9
290     do
291         [[ -r $ZONEROOT/etc/svc/volatile/repository_door ]] && break
292         sleep 5
293     done

295     if [[ $i -eq 9 && ! -r $ZONEROOT/etc/svc/volatile/repository_door ]];
296     then
297         #
298         # The zone never booted, something is wrong.
299         #
300         error "$e_nosmf"
301         error "$e_bootfail"
302         /usr/bin/rm -f $SMFTMPFILE
303         return 1
304     fi

306     insttmpfile=$(mktemp -t instsmf.XXXXXX)
307     if [[ $? == 1 || -z "$insttmpfile" ]]; then
308         error "$e_tmpfile"
309         /usr/bin/rm -f $SMFTMPFILE
310         return 1
311     fi

313     vlog "$v_rmhollowsvcs"
314     while read fmri mfst
315     do
316         # Delete the svc.
317         vlog "$v_delsvc" "$fmri"
318         echo "/usr/sbin/svccfg delete -f $fmri"
319         echo "/usr/sbin/svccfg delhash -d $mfst"
320         echo "rm -f $mfst"
321     done < $SMFTMPFILE > $ZONEROOT/tmp/smf_rm

323     /usr/sbin/zlogin -S $ZONENAME /bin/sh /tmp/smf_rm >/dev/null 2>&1

325     /usr/bin/rm -f $SMFTMPFILE

327     # Get a list of the svcs that now exist in the zone.
328     LANG=C /usr/sbin/zlogin -S $ZONENAME /usr/bin/svcs -aH | \
329     /usr/xpg4/bin/awk '{print $3}' >>$insttmpfile
329     /usr/bin/nawk '{print $3}' >>$insttmpfile

331     [[ -n $LOGFILE ]] && \
332     printf "[%s]" $(date) "${MSG_PREFIX}${v_svcsinzone}\n" >&2
333     [[ -n $LOGFILE ]] && cat $insttmpfile >&2

```

```

335 #
336 # Import ip-interface-management service in S10C, network
337 # loopback service requires ipmgmt in exclusive stack zones.
338 #
339 /usr/sbin/zlogin -S $ZONENAME /usr/sbin/svccfg import \
340 $ZONEROOT/var/svc/manifest/network/network-ipmgmt.xml
341
342 #
343 # Fix network services if shared stack.
344 #
345 if [[ "$STACK_TYPE" == "shared" ]]; then
346     vlog "$v_fixnetsvcs"
347
348     NETPHYSDEF="svc:/network/physical:default"
349     NETPHYSNWAM="svc:/network/physical:nwam"
350
351     /usr/bin/egrep -s "$NETPHYSDEF" $insttmpfile
352     if (( $? == 0 )); then
353         vlog "$v_enblsvcs" "$NETPHYSDEF"
354         /usr/sbin/zlogin -S $ZONENAME \
355             /usr/sbin/svccadm enable $NETPHYSDEF || \
356             error "$e_dissvc" "$NETPHYSDEF"
357     fi
358
359     /usr/bin/egrep -s "$NETPHYSNWAM" $insttmpfile
360     if (( $? == 0 )); then
361         vlog "$v_dissvc" "$NETPHYSNWAM"
362         /usr/sbin/zlogin -S $ZONENAME \
363             /usr/sbin/svccadm disable $NETPHYSNWAM || \
364             error "$e_enblsvcs" "$NETPHYSNWAM"
365     fi
366
367     for i in $(/usr/bin/egrep network/routing $insttmpfile)
368     do
369         # Disable the svc.
370         vlog "$v_dissvc" "$i"
371         /usr/sbin/zlogin -S $ZONENAME \
372             /usr/sbin/svccadm disable $i || \
373             error "$e_dissvc" $i
374     done
375 fi
376
377 #
378 # Disable well-known services that don't run in a zone.
379 #
380 vlog "$v_rminvalidsvcs"
381 for i in $(/usr/bin/egrep -hv "^#" \
382 /usr/lib/brand/solaris10/smf_disable.lst \
383 /etc/brand/solaris10/smf_disable.conf)
384 do
385     # Skip svcs not installed in the zone.
386     /usr/bin/egrep -s "$i:" $insttmpfile || continue
387
388     # Disable the svc.
389     vlog "$v_dissvc" "$i"
390     /usr/sbin/zlogin -S $ZONENAME /usr/sbin/svccadm disable $i || \
391     error "$e_dissvc" $i
392 done
393
394 #
395 # Since zones can't be NFS servers, disable all of the instances of
396 # the shares svc.
397 #
398 for i in $(/usr/bin/egrep network/shares/group $insttmpfile)
399 do

```

```

400     vlog "$v_dissvc" "$i"
401     /usr/sbin/zlogin -S $ZONENAME /usr/sbin/svccadm disable $i || \
402     error "$e_dissvc" $i
403 done
404
405 /usr/bin/rm -f $insttmpfile
406
407     return 0
408 }
409
410 _____unchanged_portion_omitted_____
411
412 #
413 # Zoneadmd writes a one-line index file into the zone when the zone boots,
414 # so any information about installed zones from the original system will
415 # be lost at that time. Here we'll warn the sysadmin about any pre-existing
416 # zones that they might want to clean up by hand, but we'll leave the zonepaths
417 # in place in case they're on shared storage and will be migrated to
418 # a new host.
419 #
420 warn_zones()
421 {
422     zoneconfig=$ZONEROOT/etc/zones
423
424     [[ ! -d $zoneconfig ]] && return
425
426     if [[ -h $zoneconfig/index || ! -f $zoneconfig/index ]]; then
427         error "$e_badfile" "/etc/zones/index"
428     fi
429
430     NGZ=$( /usr/xpg4/bin/awk -F: '{
431         NGZ=$( /usr/bin/nawk -F: '{
432             if (substr($1, 0, 1) == "#" || $1 == "global")
433                 continue
434
435             if ($2 == "installed")
436                 printf("%s ", $1)
437         }' $zoneconfig/index)
438
439     # Return if there are no installed zones to warn about.
440     [[ -z "$NGZ" ]] && return
441
442     log "$v_rmzones" "$NGZ"
443
444     NGZP=$( /usr/xpg4/bin/awk -F: '{
445         NGZP=$( /usr/bin/nawk -F: '{
446             if (substr($1, 0, 1) == "#" || $1 == "global")
447                 continue
448
449             if ($2 == "installed")
450                 printf("%s ", $3)
451         }' $zoneconfig/index)
452
453     log "$v_rmzonepaths"
454
455     for i in $NGZP
456     do
457         log "    %s" "$i"
458     done
459 }
460
461 #
462 # ^C Should cleanup; if the zone is running, it should try to halt it.
463 #
464 zone_is_running=0
465 trap trap_cleanup INT

```

```

494 #
495 # Parse the command line options.
496 #
497 OPT_U=
498 OPT_V=
499 OPT_M=
500 OPT_L=
501 while getopts "uvm:l:" opt
502 do
503     case "$opt" in
504         u)    OPT_U="-u";;
505         v)    OPT_V="-v";;
506         m)    MSG_PREFIX="$OPTARG"; OPT_M="-m \"$OPTARG\"";;
507         l)    LOGFILE="$OPTARG"; OPT_L="-l \"$OPTARG\"";;
508         *)    usage;;
509     esac
510 done
511 shift OPTIND-1

513 (( $# < 1 )) && usage

515 (( $# > 2 )) && usage

517 [[ -n $LOGFILE ]] && exec 2>>$LOGFILE

519 ZONENAME=$1
520 ZONEPATH=$2
521 # XXX shared/common script currently uses lower case zonename & zonepath
522 zonename="$ZONENAME"
523 zonepath="$ZONEPATH"
524 ZONEROOT=$ZONEPATH/root

526 e_badinfo=$(gettext "Failed to get '%s' zone resource")
527 e_badfile=$(gettext "Invalid '%s' file within the zone")
528 v_mkdirs=$(gettext "Creating mount points")
529 v_nonetfix=$(gettext "Cannot update /etc/hostname.{net} file")
530 v_adjust=$(gettext "Updating the image to run within a zone")
531 v_stacktype=$(gettext "Stack type '%s'")
532 v_booting=$(gettext "Booting zone to single user mode")
533 e_bootfail=$(gettext "Failed to boot zone to single user mode.")
534 e_nosmf=$(gettext "SMF repository unavailable.")
535 v_svcsinzone=$(gettext "The following SMF services are installed:")
536 v_rmhollowsvcs=$(gettext "Deleting SMF services from hollow packages")
537 v_fixnetsvcs=$(gettext "Adjusting network SMF services")
538 v_rminvalidsvcs=$(gettext "Disabling invalid SMF services")
539 v_delsvc=$(gettext "Delete SMF svc '%s'")
540 e_delsvc=$(gettext "deleting SMF svc '%s'")
541 v_enblsvc=$(gettext "Enable SMF svc '%s'")
542 e_enblsvc=$(gettext "enabling SMF svc '%s'")
543 v_disssvc=$(gettext "Disable SMF svc '%s'")
544 e_disssvc=$(gettext "disabling SMF svc '%s'")
545 e_adminf=$(gettext "Unable to create admin file")
546 v_rmpkg=$(gettext "Remove package '%s'")
547 e_rmpkg=$(gettext "removing package '%s'")
548 v_rmzones=$(gettext "The following zones in this image will be unusable: %s")
549 v_rmzonepaths=$(gettext "These zonepaths could be removed from this image:")
550 v_halting=$(gettext "Halting zone")
551 e_shutdown=$(gettext "Shutting down zone %s...")
552 e_badhalt=$(gettext "Zone halt failed")
553 v_exitgood=$(gettext "Postprocessing successful.")
554 e_exitfail=$(gettext "Postprocessing failed.")

556 #
557 # Do some validation on the paths we'll be accessing
558 #

```

```

559 safe_dir /etc
560 safe_dir /var
561 safe_dir /var/sadm
562 safe_dir /var/sadm/install
563 safe_dir /var/sadm/pkg
564 safe_opt_dir /etc/dfs
565 safe_opt_dir /etc/lu
566 safe_opt_dir /etc/zones

568 mk_zone_dirs

570 # Now do the work to update the zone.

572 # Check for zones inside of image.
573 warn_zones
574 fix_smf_pre_uoa

576 log "$v_adjust"

578 #
579 # Any errors in these functions are not considered fatal. The zone can be
580 # be fixed up manually afterwards and it may need some additional manual
581 # cleanup in any case.
582 #

584 STACK_TYPE=$(/usr/sbin/zoneadm -z $ZONENAME list -p | \
585     /usr/xpg4/bin/awk -F: '{print $7}')
585     /usr/bin/nawk -F: '{print $7}'
586 if (( $? != 0 )); then
587     error "$e_badinfo" "stacktype"
588 fi
589 vlog "$v_stacktype" "$STACK_TYPE"

591 fix_lu
592 fix_net
593 fix_nfs
594 fix_vfstab

596 vlog "$v_booting"

598 #
599 # Boot the zone so that we can do all of the SMF updates needed on the zone's
600 # repository.
601 #

603 zone_is_running=1

605 /usr/sbin/zoneadm -z $ZONENAME boot -f -- -m milestone=none
606 if (( $? != 0 )); then
607     error "$e_badboot"
608     /usr/bin/rm -f $SMFTMPFILE
609     fatal "$e_exitfail"
610 fi

612 #
613 # Remove all files and directories installed by hollow packages. Such files
614 # and directories shouldn't exist inside zones.
615 #
616 hollow_pkgs=$(mktemp -t .hollow.pkgs.XXXXXX)
617 hollow_file_list=$(mktemp $ZONEROOT/.hollow.pkgs.files.XXXXXX)
618 hollow_dir_list=$(mktemp $ZONEROOT/.hollow.pkgs.dirs.XXXXXX)
619 [ -f "$hollow_pkgs" -a -f "$hollow_file_list" -a -f "$hollow_dir_list" ] || {
620     error "$e_tmpfile"
621     rm -f $hollow_pkgs $hollow_file_list $hollow_dir_list
622     fatal "$e_exitfail"
623 }

```

```

624 for pkg_name in $ZONEROOT/var/sadm/pkg/*; do
625     grep 'SUNW_PKG_HOLLOW=true' $pkg_name/pkginfo >/dev/null 2>&1 && \
626         basename $pkg_name >>$hollow_pkgs
627 done
628 /usr/xpg4/bin/awk -v hollowpkgs=$hollow_pkgs -v filelist=$hollow_file_list \
629 /usr/bin/nawk -v hollowpkgs=$hollow_pkgs -v filelist=$hollow_file_list \
629     -v dirlist=$hollow_dir_list '
630     BEGIN {
631         while (getline p <hollowpkgs > 0)
632             pkgs[p] = 1;
633         close(hollowpkgs);
634     }
635
636     # fld is the field where the pkg names begin.
637     # nm is the file/dir entry name.
638     if ($2 == "f") {
639         fld=10;
640         nm=$1;
641     } else if ($2 == "d") {
642         fld=7;
643         nm=$1;
644     } else if ($2 == "s" || $2 == "l") {
645         fld=4;
646         split($1, a, "=");
647         nm=a[1];
648     } else {
649         next;
650     }
651
652     # Determine whether the file or directory is delivered by any
653     # non-hollow packages. Files and directories can be
654     # delivered by multiple pkgs. The file or directory should only
655     # be removed if it is only delivered by hollow packages.
656     for (i = fld; i <= NF; i++) {
657         if (pkgs[get_pkg_name($i)] != 1) {
658             # We encountered a non-hollow package. Skip
659             # this entry.
660             next;
661         }
662     }
663
664     # The file or directory is only delivered by hollow packages.
665     # Mark it for removal.
666     if (fld != 7)
667         print nm >>filelist
668     else
669         print nm >>dirlist
670 }
671
672 # Get the clean pkg name from the fld entry.
673 function get_pkg_name(fld) {
674     # Remove any pkg control prefix (e.g. *, !)
675     first = substr(fld, 1, 1)
676     if (match(first, /[A-Za-z]/)) {
677         pname = fld
678     } else {
679         pname = substr(fld, 2)
680     }
681
682     # Then remove any class action script name
683     pos = index(pname, ":")
684     if (pos != 0)
685         pname = substr(pname, 1, pos - 1)
686     return (pname)
687 }
688 ' $ZONEROOT/var/sadm/install/contents

```

```

689 /usr/sbin/zlogin -S $ZONENAME "cat /$(basename $hollow_file_list) | xargs rm -f"
690 /usr/sbin/zlogin -S $ZONENAME "sort -r /$(basename $hollow_dir_list) | \
691     xargs rmdir >/dev/null 2>&1"
692 rm -f $hollow_pkgs $hollow_file_list $hollow_dir_list
693
694 # cleanup SMF services
695 fix_smf || failed=1
696
697 # remove invalid pkgs
698 [[ -z $failed ]] && rm_pkgs
699
700 if [[ -z $failed && -n $OPT_U ]]; then
701     vlog "$v_unconfig"
702
703     sysunconfig_zone
704     if (( $? != 0 )); then
705         failed=1
706     fi
707 fi
708
709 vlog "$v_halting"
710 /usr/sbin/zoneadm -z $ZONENAME halt
711 if (( $? != 0 )); then
712     error "$e_badhalt"
713     failed=1
714 fi
715 zone_is_running=0
716
717 if [[ -n $failed ]]; then
718     fatal "$e_exitfail"
719 fi
720
721 vlog "$v_exitgood"
722 exit 0

```

new/usr/src/lib/libbssm/mkhdr.sh

1

```
*****
2789 Sun May 26 18:23:23 2013
new/usr/src/lib/libbssm/mkhdr.sh
XXX Remove awk(1)
*****
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 # Copyright (c) 1992, 2010, Oracle and/or its affiliates. All rights reserved.
23 #
24 #
25 # Automatically generate the audit_uevents.h header file.
26 #
27 DATABASE=audit_event.txt
28 HEADER_FILE=audit_uevents.h
29 CR_YEAR='/usr/bin/date +%Y'

31 cat <<EOF > $HEADER_FILE
32 /*
33 * CDDL HEADER START
34 *
35 * The contents of this file are subject to the terms of the
36 * Common Development and Distribution License (the "License").
37 * You may not use this file except in compliance with the License.
38 *
39 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
40 * or http://www.opensolaris.org/os/licensing.
41 * See the License for the specific language governing permissions
42 * and limitations under the License.
43 *
44 * When distributing Covered Code, include this CDDL HEADER in each
45 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
46 * If applicable, add the following below this CDDL HEADER, with the
47 * fields enclosed by brackets "[]" replaced with your own identifying
48 * information: Portions Copyright [yyyy] [name of copyright owner]
49 *
50 * CDDL HEADER END
51 */
52 /*
53 * Copyright (c) 1992, $CR_YEAR, Oracle and/or its affiliates. All rights reserv
54 */

56 #ifndef _BSM_AUDIT_UEVENTS_H
57 #define _BSM_AUDIT_UEVENTS_H

59 EOF

61 cat <<EOF >> $HEADER_FILE
```

new/usr/src/lib/libbssm/mkhdr.sh

2

```
63 /*
64 * User level audit event numbers.
65 *
66 * 0 Reserved as an invalid event number.
67 * 1 - 2047 Reserved for the Solaris Kernel events.
68 * 2048 - 32767 Reserved for the Solaris TCB programs.
69 * 32768 - 65535 Available for third party TCB applications.
70 *
71 */

73 #ifdef __cplusplus
74 extern "C" {
75 #endif

77 EOF

79 /usr/xpg4/bin/awk -F: '{if ((NF == 4) && substr($1,0,1) != "#")
79 awk -F: '{if ((NF == 4) && substr($1,0,1) != "#")
80 if ($1 >= 2048) {
81 tlen = length($2);

83 printf("#define\t%s\t", $2)
84 if (tlen < 8)
85 printf("\t");
86 if (tlen < 16)
87 printf("\t");
88 if (tlen < 24)
89 printf("\t");
90 printf("%s\n", $1);
91 }
92 }' \
93 < $DATABASE >> $HEADER_FILE

95 cat <<EOF >> $HEADER_FILE

97 #ifdef __cplusplus
98 }

_____unchanged_portion_omitted_____
```



```

*****
15287 Sun May 26 18:23:24 2013
new/usr/src/lib/libsecdb/common/i.rbac
XXX Remove awk(1)
*****
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 # i.rbac
23 #
24 # Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
25 #
26 # class action script for "rbac" class files
27 # installed by pkgadd
28 #
29 # Files in "rbac" class:
30 #
31 # /etc/security/{prof_attr,exec_attr,auth_attr}
32 # /etc/user_attr
33 #
34 # Allowable exit codes
35 #
36 # 0 - success
37 # 2 - warning or possible error condition. Installation continues. A warning
38 # message is displayed at the time of completion.
39 #

41 umask 022

43 tmp_dir=${TMPDIR:-/tmp}

45 PATH="/usr/bin:/usr/sbin:${PATH}"
46 export PATH

48 basename_cmd=basename
49 cp_cmd=cp
50 egrep_cmd=egrep
51 mv_cmd=mv
52 awk_cmd=/usr/xpg4/bin/awk
53 rm_cmd=rm
54 sed_cmd=sed
55 sort_cmd=sort

57 # $1 is the type
58 # $2 is the "old/existing file"
59 # $3 is the "new (to be merged)" file
60 # $4 is the output file

```

```

61 # returns 0 on success
62 # returns 2 on failure if awk fails with non-zero exit status
62 # returns 2 on failure if nawk fails with non-zero exit status
63 #
64 dbmerge() {
65 #
66 # Remove the ident lines.
67 #
68     ${egrep_cmd} -v '^#[pragma      ]*ident' $2 > $4.old 2>/dev/null
69 #
70 # If the new file has a Sun copyright, remove the Sun copyright from the old
71 # file.
72 #
73     newcr='${egrep_cmd} '^# Copyright.*Sun Microsystems, Inc.' $3 \
74     2>/dev/null`
75     if [ -n "${newcr}" ]; then
76         $sed_cmd -e '/^# Copyright.*Sun Microsystems, Inc./d' \
77             -e '/^# All rights reserved./d' \
78             -e '/^# Use is subject to license terms./d' \
79             $4.old > $4.$$ 2>/dev/null
80         $mv_cmd $4.$$ $4.old
81     fi
82 #
83 # If the new file has an Oracle copyright, remove both the Sun and Oracle
84 # copyrights from the old file.
85 #
86     oracle_cr='${egrep_cmd} '^# Copyright.*Oracle and/or its affiliates.' \
87     $3 2>/dev/null`
88     if [ -n "${oracle_cr}" ]; then
89         $sed_cmd -e '/^# Copyright.*Sun Microsystems, Inc./d' \
90             -e '/^# All rights reserved./d' \
91             -e '/^# Use is subject to license terms./d' \
92             -e '/^# Copyright.*Oracle and/or its affiliates./d' \
93             $4.old > $4.$$ 2>/dev/null
94         $mv_cmd $4.$$ $4.old
95     fi
96 #
97 # If the new file has the CDDL, remove it from the old file.
98 #
99     newcr='${egrep_cmd} '^# CDDL HEADER START' $3 2>/dev/null`
100    if [ -n "${newcr}" ]; then
101        $sed_cmd -e '/^# CDDL HEADER START/,/^# CDDL HEADER END/d' \
102            $4.old > $4.$$ 2>/dev/null
103        $mv_cmd $4.$$ $4.old
104    fi
105 #
106 # Remove empty lines and multiple instances of these comments:
107 #
108    $sed_cmd -e '/^# \\/etc\/security\/exec_attr/d' -e '/^#$/d' \
109        -e '/^# execution attributes for profiles./d' \
110        -e '/^# See exec_attr(4)/d' \
111        -e '/^# \\/etc\/user_attr/d' \
112        -e '/^# user attributes. see user_attr(4)/d' \
113        -e '/^# \\/etc\/security\/prof_attr/d' \
114        -e '/^# profiles attributes. see prof_attr(4)/d' \
115        -e '/^# See prof_attr(4)/d' \
116        -e '/^# \\/etc\/security\/auth_attr/d' \
117        -e '/^# authorizations. see auth_attr(4)/d' \
118        -e '/^# authorization attributes. see auth_attr(4)/d' \
119        $4.old > $4.$$
120    $mv_cmd $4.$$ $4.old
121 #
122 # Retain old and new header comments.
123 #
124    $sed_cmd -n -e '/^[^#]/,$d' -e '/^##/,,$d' -e p $4.old > $4
125    $rm_cmd $4.old

```

```

126     $sed_cmd -n -e '/^[^#]//,$d' -e '/^##/,,$d' -e p $3 >> $4
127 #
128 # If the output file now has both Sun and Oracle copyrights, remove
129 # the Sun copyright.
130 #
131     sun_cr='${egrep_cmd}' '^# Copyright.*Sun Microsystems, Inc.' \
132     $4 2>/dev/null`
133     oracle_cr='${egrep_cmd}' '^# Copyright.*Oracle and/or its affiliates.' \
134     $4 2>/dev/null`
135     if [ -n "${sun_cr}" ] && [ -n "${oracle_cr}" ]; then
136         $sed_cmd -e '/^# Copyright.*Sun Microsystems, Inc./d' \
137             -e '/^# All rights reserved./d' \
138             -e '/^# Use is subject to license terms./d' \
139             $4 > $4.$$ 2>/dev/null
140         $mv_cmd $4.$$ $4
141     fi
142 #
143 # Handle line continuations (trailing \)
144 #
145     $sed_cmd \
146     -e '/\\$/ {N;s/\\n//;}' -e '/\\$/ {N;s/\\n//;}' \
147     -e '/\\$/ {N;s/\\n//;}' -e '/\\$/ {N;s/\\n//;}' \
148     -e '/\\$/ {N;s/\\n//;}' -e '/\\$/ {N;s/\\n//;}' \
149     $2 > $4.old
150     $sed_cmd \
151     -e '/\\$/ {N;s/\\n//;}' -e '/\\$/ {N;s/\\n//;}' \
152     -e '/\\$/ {N;s/\\n//;}' -e '/\\$/ {N;s/\\n//;}' \
153     -e '/\\$/ {N;s/\\n//;}' -e '/\\$/ {N;s/\\n//;}' \
154     $3 > $4.new
155 #
156 # The awk script below processes the old and new files using up to
157 # three passes. If the old file is empty, only the final pass over
158 # the new file is required.
159 #
160     if [ -s $4.old ]; then
161         awk_pass1=$4.old
162         awk_pass2=$4.new
163         awk_pass3=$4.new
164         nawk_pass1=$4.old
165         nawk_pass2=$4.new
166         nawk_pass3=$4.new
167     else
168         awk_pass1=
169         awk_pass2=
170         awk_pass3=$4.new
171         nawk_pass1=
172         nawk_pass2=
173         nawk_pass3=$4.new
174     fi
175 #
176 # dbmerge type=[auth|prof|user|exec] [ old-file new-file ] new-file
177 #
178 # Merge two versions of an RBAC database file. The output
179 # consists of the lines from the new-file, while preserving
180 # user customizations in the old-file.
181 #
182 # Entries in the new-file replace corresponding entries in the
183 # old-file, except as follows: For exec_attr, all old entries
184 # for profiles contained in the new-file are discarded. For
185 # user_attr, the "root" entry from the old-file is retained,
186 # and new keywords from the new-file are merged into it.

```

```

184 # Records with the same key field(s) are merged, so that the
185 # keyword/value section of each output record contains the union
186 # of the keywords found in all input records with the same key
187 # field(s). For selected multi-value keywords [1] the values from
188 # the new-file are merged with retained values from the old-file.
189 # Otherwise, the value for each keyword is the final value found
190 # in the new-file, except for keywords in the user_attr entry for
191 # "root" where values from the old-file are always retained.
192 #
193 # [1] The following file type and keyword combinations are merged:
194 #     prof_attr: auths, profiles, privs
195 #     user_attr: auths, profiles, roles
196 #
197 # The output is run through sort except for the comments
198 # which will appear first in the output.
199 #
200 #
201     $awk_cmd '
202     $nawk_cmd '
203 # This script may be invoked with up to three file names. Each file
204 # name corresponds to a separate processing pass. The passes are
205 # defined as follows:
206 #
207 # Pass 1: Read existing data.
208 # Data from the old-file is read into memory.
209 #
210 # Pass 2: Remove obsolete data.
211 # Discard any data from the old-file that is part of profiles that
212 # are also in the new-file. (As a special case, the user_attr entry
213 # for 'root' is always retained.)
214 #
215 # Pass 3: Merge new data.
216 # Data from the new-file is merged with the remaining old-file data.
217 # (As a special case, exec_attr entries are replaced, not merged.)
218 #
219 BEGIN {
220     # The variable 'pass' specifies which type of processing to perform.
221     # When processing only one file, skip passes 1 and 2.
222     if (ARGC == 3)
223         pass += 2;
224 #
225 # The array 'keyword_behavior' specifies the special treatment of
226 # [type, keyword] combinations subject to value merging.
227 keyword_behavior["prof", "auths"] = "merge";
228 keyword_behavior["prof", "profiles"] = "merge";
229 keyword_behavior["prof", "privs"] = "merge";
230 keyword_behavior["user", "auths"] = "merge";
231 keyword_behavior["user", "profiles"] = "merge";
232 keyword_behavior["user", "roles"] = "merge";
233 #
234     FS=":"
235 }
236 #
237 # When FNR (current file record number) is 1 it indicates that awk
238 # is starting to read the next file specified on its command line,
239 # and is beginning the next processing pass.
240 FNR == 1 {
241     pass++;
242 }
243 #
244 /^#/ || /^${ {
245     next;
246 }

```

```

248 {
249     # For each input line, awk automatically assigns the complete
250     # For each input line, awk automatically assigns the complete
251     # line to $0 and also splits the line at field separators and
252     # assigns each field to a variable $1..$n. Assignment to $0
253     # re-splits the line into the field variables. Conversely,
254     # assignment to a variable $1..$n will cause $0 to be recomputed
255     # assignment to a variable $1..$n will cause $0 to be recomputed
256     # from the field variable values.
257     #
258     # This code adds awareness of escaped field separators by using
259     # a custom function to split the line into a temporary array.
260     # It assigns the empty string to $0 to clear any excess field
261     # variables, and assigns the desired elements of the temporary
262     # array back to the field variables $1..$7.
263     #
264     # Subsequent code must not assign directly to $0 or the fields
265     # will be re-split without regard to escaped field separators.
266     split_escape($0, f, ":");
267     $0 = "";
268     $1 = f[1];
269     $2 = f[2];
270     $3 = f[3];
271     $4 = f[4];
272     $5 = f[5];
273     $6 = f[6];
274     $7 = f[7];
275 }

```

unchanged_portion_omitted_

```

423 # This function is similar to the awk built-in split() function,
424 # This function is similar to the nawk built-in split() function,
425 # except that a "\" character may be used to escape any subsequent
426 # character, so that the escaped character will not be treated as a
427 # field separator or as part of a field separator regular expression.
428 # The "\" characters will remain in the elements of the output array
429 # variable upon completion.
430 function split_escape(str, list, fs, cnt, saved, sep)
431 {
432     # default to global FS
433     if (fs == "")
434         fs = FS;
435     # initialize empty list, cnt, saved
436     split("", list, " ");
437     cnt = 0;
438     saved = "";
439     # track whether last token was a field separator
440     sep = 0;
441     # nonzero str length indicates more string left to scan
442     while (length(str)) {
443         if (match(str, fs) == 1) {
444             # field separator, terminates current field
445             list[++cnt] = saved;
446             saved = "";
447             str = substr(str, RLENGTH + 1);
448             sep = 1;
449         } else if (substr(str, 1, 1) == "\\") {
450             # escaped character
451             saved = saved substr(str, 1, 2);
452             str = substr(str, 3);
453             sep = 0;
454         } else {
455             # regular character
456             saved = saved substr(str, 1, 1);
457             str = substr(str, 2);

```

```

457         sep = 0;
458     }
459 }
460 # if required, append final field to list
461 if (sep || length(saved))
462     list[++cnt] = saved;
463
464     return cnt;
465 }
466
467 unchanged_portion_omitted_
468 type=$1 $awk_pass1 $awk_pass2 $awk_pass3 > $4.unsorted
469 type=$1 $nawk_pass1 $nawk_pass2 $nawk_pass3 > $4.unsorted
470 rc=?
471 $sort_cmd < $4.unsorted >> $4
472 return $rc
473 }
474
475 unchanged_portion_omitted_

```

new/usr/src/lib/libshell/common/tests/quoting.sh

1

13433 Sun May 26 18:23:24 2013

new/usr/src/lib/libshell/common/tests/quoting.sh

XXX Remove `nawk(1)`

unchanged portion omitted

```
330 [[ $x == x ]] || err_exit 'assignment ${x:=`...`} failed'
331 [[ ${SHELL} -c 'print a[' == 'a[' ]] || err_exit "unbalanced '[' in command arg
332 $SHELL -c $'false && (( `wc -l /dev/null | /usr/xpg4/bin/awk \'{print $1}\'' > 2
332 $SHELL -c $'false && (( `wc -l /dev/null | nawk \'{print $1}\'' > 2 )) && true;;
333 { $SHELL -c '( ( 1: "{ }" ) )' ; } 2> /dev/null || err_exit 'problem with ` ins
334 varname=foobarx
335 x=`print "'\${varname}'"``
336 [[ $x == "$foobarx" ]] || err_exit '$'\$\` not handled correctly inside ``

338 copy1=5 copynum=1
339 foo=`eval echo "${copy$copynum}-0}"``
340 [[ $foo == "$copy1" ]] || err_exit '$"... " not being ignored inside ``

342 exit $((Errors))
```

new/usr/src/lib/libshell/common/tests/sun_solaris_builtin_sum.sh 1

```

*****
5604 Sun May 26 18:23:29 2013
new/usr/src/lib/libshell/common/tests/sun_solaris_builtin_sum.sh
XXX Remove awk(1)
*****
unchanged portion omitted
38 alias err_exit='err_exit $LINENO'

40 set -o nounset
41 Command=${0##*/}
42 integer Errors=0

45 typeset x

47 builtin sum || err_exit "sum builtin not found"

49 # Basic tests
50 x="$(print 'hello' | /usr/bin/sum)" || err_exit "/usr/bin/sum pipe failed."
51 [[ "$x" == "542 1" ]] || err_exit "print 'hello' | /usr/bin/sum did not return 5"
52 x="$(print 'hello' | sum)" || err_exit "sum builtin pipe failed."
53 [[ "$x" == "542 1" ]] || err_exit "print 'hello' | sum builtin did not return 54"
54 x="$(print 'hello' | sum -x md5)" || err_exit "sum md5 builtin pipe failed."
55 [[ "$x" == "b1946ac92492d2347c6235b4d2611184" ]] || err_exit "print 'hello' | su"
56 x="$(print 'hello' | sum -x shal)" || err_exit "sum shal builtin pipe failed."
57 [[ "$x" == "f572d396fae9206628714fb2ce00f72e94f2258f" ]] || err_exit "print 'hel"
58 x="$(print 'hello' | sum -x sha256)" || err_exit "sum sha256 builtin pipe failed"
59 [[ "$x" == "5891b5b522d5df086d0ff0b110fbd9d21bb4fc7163af34d08286a2e846f6be03" ]]
60 x="$(print 'hello' | sum -x sha512)" || err_exit "sum sha512 builtin pipe failed"
61 [[ "$x" == "e7c22b994c59d9cf2b48e549ble24666636045930d3da7c1acb299dlc3b7f931f94a"
62 # note that Solaris /usr/bin/cksum outputs '$3015617425\t6' (which may be a bug
63 x="$(print 'hello' | sum -x cksum)" || err_exit "sum cksum builtin pipe failed."
64 [[ "$x" == '$3015617425 6' ]] || err_exit "print 'hello' | sum cksum builtin did

66 [[ "$(print 'hello' | /usr/bin/sum)" == "$(print 'hello' | sum)" ]] || e
67 [[ "$(print 'fish' | /usr/bin/sum)" == "$(print 'fish' | sum)" ]] || e
68 [[ "$(print '12345' | /usr/bin/sum)" == "$(print '12345' | sum)" ]] || e
69 [[ "$(print '\n\r\n \v' | /usr/bin/sum)" == "$(print '\n\r\n \v' | sum)" ]] || e

71 # Test some binary files...
72 x="/usr/bin/ls"
73 [[ "$(cat "$x" | /usr/bin/sum)" == "$(cat "$x" | sum)" ]] || err_exit "pip
74 [[ "$(/usr/bin/sum "$x")" == "$(sum "$x")" ]] || err_exit "fil

76 x="/usr/bin/chmod"
77 [[ "$(cat "$x" | /usr/bin/sum)" == "$(cat "$x" | sum)" ]] || err_exit "pip
78 [[ "$(/usr/bin/sum "$x")" == "$(sum "$x")" ]] || err_exit "fil

80 x="/usr/bin/tee"
81 [[ "$(cat "$x" | /usr/bin/sum)" == "$(cat "$x" | sum)" ]] || err_exit "pip
82 [[ "$(/usr/bin/sum "$x")" == "$(sum "$x")" ]] || err_exit "fil

84 x="/usr/bin/grep"
85 [[ "$(cat "$x" | /usr/bin/sum)" == "$(cat "$x" | sum)" ]] || err_exit "pip
86 [[ "$(/usr/bin/sum "$x")" == "$(sum "$x")" ]] || err_exit "fil

88 x="/usr/bin/egrep"
89 [[ "$(cat "$x" | /usr/bin/sum)" == "$(cat "$x" | sum)" ]] || err_exit "pip
90 [[ "$(/usr/bin/sum "$x")" == "$(sum "$x")" ]] || err_exit "fil

92 x="/usr/bin/awk"
93 [[ "$(cat "$x" | /usr/bin/sum)" == "$(cat "$x" | sum)" ]] || err_exit "pip
94 [[ "$(/usr/bin/sum "$x")" == "$(sum "$x")" ]] || err_exit "fil

96 x="/usr/xpg4/bin/awk"
96 x="/usr/bin/nawk"

```

new/usr/src/lib/libshell/common/tests/sun_solaris_builtin_sum.sh 2

```

97 [[ "$(cat "$x" | /usr/bin/sum)" == "$(cat "$x" | sum)" ]] || err_exit "pip
98 [[ "$(/usr/bin/sum "$x")" == "$(sum "$x")" ]] || err_exit "fil

100 x="/usr/bin/ksh"
101 [[ "$(cat "$x" | /usr/bin/sum)" == "$(cat "$x" | sum)" ]] || err_exit "pip
102 [[ "$(/usr/bin/sum "$x")" == "$(sum "$x")" ]] || err_exit "fil

104 x="/usr/bin/sh"
105 [[ "$(cat "$x" | /usr/bin/sum)" == "$(cat "$x" | sum)" ]] || err_exit "pip
106 [[ "$(/usr/bin/sum "$x")" == "$(sum "$x")" ]] || err_exit "fil

109 # tests done
110 exit $((Errors))

```

```

*****
2588 Sun May 26 18:23:30 2013
new/usr/src/lib/libmbfs/Makefile.com
XXX Remove awk(1)
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #

22 #
23 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #

27 #
28 # lib/libmbfs/Makefile.com
29 #

31 LIBRARY=      libmbfs.a
32 VERS=         .1

34 # leaving out: kiconv.o

36 OBJ_LIB=\
37     acl_api.o \
38     acl_print.o \
39     charsets.o \
40     cfopt.o \
41     connect.o \
42     crypt.o \
43     ctx.o \
44     derparse.o \
45     file.o \
46     findvc.o \
47     getaddr.o \
48     iod_cl.o \
49     iod_wk.o \
50     keychain.o \
51     krb5ssp.o \
52     mbuf.o \
53     nb.o \
54     nb_name.o \
55     nb_net.o \
56     nb_ssn.o \
57     nbns_rq.o \
58     negprot.o \
59     netshareenum.o \
60     newvc.o \
61     nls.o \

```

```

62     ntlm.o \
63     ntlmssp.o \
64     print.o \
65     rap.o \
66     rcfile.o \
67     rq.o \
68     signing.o \
69     spnego.o \
70     spnego.o \
71     ssnsetup.o \
72     ssp.o \
73     subr.o \
74     ui-sun.o \
75     utf_str.o

77 OBJ_CMN= smbfs_ntacl.o

79 OBJECTS= $(OBJ_LIB) $(OBJ_CMN)

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

83 LIBS =      $(DYNLIB) $(LINTLIB)

85 SRCDIR=     ../smb
86 CMNDIR=     $(SRC)/common/smbclnt

88 SRCS=       $(OBJ_LIB:%.o=$(SRCDIR)/%.c) \
89             $(OBJ_CMN:%.o=$(CMNDIR)/%.c)

91 $(LINTLIB) := SRCS = $(SRCDIR)/$(LINTSRC)

93 C99MODE=    $(C99_ENABLE)

95 LDLIBS += -lsocket -lnsl -lc -lmd -lpkcs11 -lkrb5 -lsec -lidmap

97 # normal warnings...
98 CFLAGS +=  $(CVERBOSE)

100 CERRWARN += -_gcc=-Wno-uninitialized
101 CERRWARN += -_gcc=-Wno-unused-variable

103 CPPFLAGS += -D__EXTENSIONS__ -D_REENTRANT -DMIA \
104             -I$(SRCDIR) -I.. \
105             -I$(SRC)/uts/common \
106             -I$(SRC)/common/smbclnt

108 # Debugging
109 ${NOT_RELEASE_BUILD} CPPFLAGS += -DDEBUG

111 # uncomment these for dbx debugging
112 #COPTFLAG = -g
113 #CTF_FLAGS =
114 #CTFCONVERT_O=
115 #CTFMERGE_LIB=

117 # Filter out the less important lint.
118 # See lgrep.awk
119 LGREP = /usr/xpg4/bin/awk -f $(SRCDIR)/lgrep.awk
120 LTAIL += 2>&1 | $(LGREP)

122 all:      $(LIBS)

124 lint:    lintcheck_t

126 include ../../Makefile.targ

```

```
128 lintcheck_t: $$SRCS
129     $(LINT.c) $(LINTCHECKFLAGS) $(SRCS) $(LDLIBS) $(LTAIL)

131 objs/%.o pics/%.o: $(CMNDIR)/%.c
132     $(COMPILE.c) -o $@ $<
133     $(POST_PROCESS_O)

135 .KEEP_STATE:
```


new/usr/src/lib/libmbfs/smb/lgrep.awk

1

```
*****
2573 Sun May 26 18:23:30 2013
new/usr/src/lib/libmbfs/smb/lgrep.awk
XXX Remove nawk(1)
*****
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 # Copyright 2012 Milan Jurik. All rights reserved.
27 #
28 #
29 # This is a "lint tail" that removes all the
30 # uninteresting lines from our lint output.
31 # It's awk because sed doesn't do (a|b).
32 # It's nawk because sed doesn't do (a|b).
33 # Also comments are easier here.
34 #
35 # There's no lintlib for krb5 yet (CR 6911968)
36 # Warning: -lkrb5 not found/ { next; }
37 # Warning: library -lkrb5 not found/ { next; }
38 # Kill noise from xti.h with _XOPEN_SOURCE vs not. (CR 6911717)
39 # _xti.* .E_INCONS_ARG_DECL2./ { next; }
40 # _xti.* .E_INCONS_ARG_USED2./ { next; }
41 # _xti.* .E_INCONS_VAL_TYPE_DECL2./ { next; }
42 #
43 # This is third-party code we'd rather not "fix"
44 # /\spnego.c.* .E_STMT_NOT_REACHED./ { next; }
45 #
46 # The mb_put/md_get functions are intentionally used both
47 # with and without return value checks. Not a concern.
48 # mb_put.* .E_FUNC_RET_[A-Z]*_IGNOR/ { next; }
49 # md_get.* .E_FUNC_RET_[A-Z]*_IGNOR/ { next; }
50 #
51 # The rc_get* functions clear the out arg even on failure,
52 # so most callers don't need to check the return value.
53 # rc_get[a-z]* .E_FUNC_RET_[A-Z]*_IGNOR/ { next; }
54 #
55 # These have uninteresting return values, usually ignored.
56 # (n|sm)b_ctx_readrcsection .E_FUNC_RET_[A-Z]*_IGNOR/ { next; }
57 # nls_str(lower|upper) .E_FUNC_RET_[A-Z]*_IGNOR/ { next; }
58 # rc_(close|freesect) .E_FUNC_RET_[A-Z]*_IGNOR/ { next; }
59 #
60 # Other functions for which we often ignore return values.
```

new/usr/src/lib/libmbfs/smb/lgrep.awk

2

```
61 # [a-z]*close .E_FUNC_RET_[A-Z]*_IGNOR/ { next; }
62 # [a-z]*flush .E_FUNC_RET_[A-Z]*_IGNOR/ { next; }
63 # [a-z]*printf .E_FUNC_RET_[A-Z]*_IGNOR/ { next; }
64 # mem(cpy|move|set) .E_FUNC_RET_[A-Z]*_IGNOR/ { next; }
65 # mutex.* .E_FUNC_RET_[A-Z]*_IGNOR/ { next; }
66 # str[ln]?(cat|cpy) .E_FUNC_RET_[A-Z]*_IGNOR/ { next; }
67 #
68 { print; }
```

new/usr/src/lib/libxcurses/Makefile

1

```
*****
2407 Sun May 26 18:23:30 2013
new/usr/src/lib/libxcurses/Makefile
XXX Remove awk(1)
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2006 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #

26 include ../Makefile.lib

28 SUBDIRS = $(MACH)
29 $(BUILD64)SUBDIRS += $(MACH64)

31 all := TARGET= all
32 install := TARGET= install
33 clean := TARGET= clean
34 clobber := TARGET= clobber
35 lint := TARGET= lint

37 # capability NAMES generated source
38 NAMES= \
39 src/libc/xcurse/boolcode.c src/libc/xcurse/boolfnam.c \
40 src/libc/xcurse/boolname.c src/libc/xcurse/numcode.c \
41 src/libc/xcurse/numfnam.c src/libc/xcurse/numname.c \
42 src/libc/xcurse/strcode.c src/libc/xcurse/strfnam.c \
43 src/libc/xcurse/strname.c

45 GENERATED= \
46 $(NAMES) src/libc/xcurse/keyindex.c h/term.h

48 # definitions for install_h target
49 HDRS= h/term.h h/curse.h h/unctrl.h

51 # extra clean-up work
52 CLOBBERFILES= $(GENERATED)

54 .KEEP_STATE:

56 LIBS= $(DYNLIB) $(LINTLIB)

58 all: $(GENERATED) .WAIT $(SUBDIRS)

60 install: all .WAIT $(SUBDIRS)
```

new/usr/src/lib/libxcurses/Makefile

2

```
62 clean clobber: $(SUBDIRS)
63 $(RM) $(GENERATED)

65 lint: $(SUBDIRS)

67 # still need to implement cstyle check of term.h
68 # do this when we go through and make this library cstyle clean.
69 check: $(HDRS)

71 $(SUBDIRS): $(GENERATED)
72 @cd $@; pwd; $(MAKE) $(TARGET)

74 # derived C files
75 # mknames.awk defines functions so needs awk
76 $(NAMES) : src/libc/xcurse/mknames.awk src/libc/xcurse/caps
77 $(RM) $(NAMES)
78 cd src/libc/xcurse; $(AWK) -f mknames.awk caps
cd src/libc/xcurse; $(NAWK) -f mknames.awk caps

79 h/term.h : src/libc/xcurse/mkterm.awk src/libc/xcurse/caps
80 $(RM) $@
81 cd src/libc/xcurse; $(AWK) -f mkterm.awk caps > term.h
82 cd src/libc/xcurse; $(NAWK) -f mkterm.awk caps > term.h
82 $(MV) src/libc/xcurse/term.h h/term.h

84 src/libc/xcurse/keyindex.c : src/libc/xcurse/mkkey.awk src/libc/xcurse/caps
85 $(RM) $@
86 cd src/libc/xcurse; $(AWK) -f mkkey.awk caps > keyindex.c
87 cd src/libc/xcurse; $(NAWK) -f mkkey.awk caps > keyindex.c
```

new/usr/src/lib/libxcurses2/Makefile

1

```
*****
2577 Sun May 26 18:23:31 2013
new/usr/src/lib/libxcurses2/Makefile
XXX Remove awk(1)
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2006 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #

26 include ../Makefile.lib

28 SUBDIRS = $(MACH)
29 $(BUILD64)SUBDIRS += $(MACH64)

31 all := TARGET= all
32 install := TARGET= install
33 clean := TARGET= clean
34 clobber := TARGET= clobber
35 lint := TARGET= lint

37 # capability NAMES generated source
38 NAMES= \
39 src/libc/xcurses/boolcode.c src/libc/xcurses/boolfnam.c \
40 src/libc/xcurses/boolname.c src/libc/xcurses/numcode.c \
41 src/libc/xcurses/numfnam.c src/libc/xcurses/numname.c \
42 src/libc/xcurses/strcode.c src/libc/xcurses/strfnam.c \
43 src/libc/xcurses/strname.c

45 GENERATED= \
46 $(NAMES) src/libc/xcurses/keyindex.c h/term.h

48 # definitions for install_h target
49 HDRS= h/term.h h/curses.h h/ucntrl.h
50 ROOTHDRDIR= $(ROOT)/usr/xpg4/include
51 ROOTHDRS= $(HDRS:h/%=$(ROOTHDRDIR)/%)

53 # install rule for install_h target
54 $(ROOTHDRDIR)/%: h/%
55 $(INS.file)

57 # extra clean-up work
58 CLOBBERFILES= $(GENERATED)

60 .KEEP_STATE:
```

new/usr/src/lib/libxcurses2/Makefile

2

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

64 all: $(GENERATED) .WAIT $(SUBDIRS)

66 install: all .WAIT $(SUBDIRS)

68 clean clobber: $(SUBDIRS)
69 $(RM) $(GENERATED)

71 lint: $(SUBDIRS)

73 # still need to implement cstyle check of term.h
74 # do this when we go through and make this library cstyle clean.
75 check: $(HDRS)

77 install_h: $(ROOTHDRS)

79 $(SUBDIRS): $(GENERATED)
80 @cd $@; pwd; $(MAKE) $(TARGET)

82 # derived C files
83 # mknames.awk defines functions so needs awk
84 $(NAMES) : src/libc/xcurses/mknames.awk src/libc/xcurses/caps
85 $(RM) $(NAMES)
86 cd src/libc/xcurses; $(AWK) -f mknames.awk caps
cd src/libc/xcurses; $(NAWK) -f mknames.awk caps

87 h/term.h : src/libc/xcurses/mkterm.awk src/libc/xcurses/caps
88 $(RM) $@
89 cd src/libc/xcurses; $(AWK) -f mkterm.awk caps > term.h
90 cd src/libc/xcurses; $(NAWK) -f mkterm.awk caps > term.h
91 $(MV) src/libc/xcurses/term.h h/term.h

92 src/libc/xcurses/keyindex.c : src/libc/xcurses/mkkey.awk src/libc/xcurses/caps
93 $(RM) $@
94 cd src/libc/xcurses; $(AWK) -f mkkey.awk caps > keyindex.c
95 cd src/libc/xcurses; $(NAWK) -f mkkey.awk caps > keyindex.c
```

new/usr/src/lib/lvm/libmeta/Makefile.com

1

5410 Sun May 26 18:23:31 2013

new/usr/src/lib/lvm/libmeta/Makefile.com

XXX Remove awk(1)

```
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 LIBRARY= libmeta.a
27 VERS= .1
28 COMMON = $(SRC)/common/lvm
```

```
30 CMN_OBJS = md_crc.o md_convert.o md_revchk.o
```

```
32 DERIVED_OBJS = \
33     mdiox_xdr.o \
34     meta_basic_xdr.o \
35     metad_clnt.o \
36     metad_xdr.o \
37     metamed_clnt.o \
38     metamed_xdr.o \
39     metamhd_clnt.o \
40     metamhd_xdr.o \
41     mdmn_commd_xdr.o \
42     mhdx_xdr.o
```

```
44 LOCAL_OBJS= \
45     metad_svc_stubs.o \
46     meta_admin.o \
47     meta_attach.o \
48     meta_db.o \
49     meta_db_balance.o \
50     meta_devadm.o \
51     meta_devstamp.o \
52     meta_error.o \
53     meta_getdevs.o \
54     meta_hotspares.o \
55     meta_import.o \
56     meta_init.o \
57     meta_lib_prv.o \
58     meta_mdcf.o \
59     meta_med_err.o \
```

new/usr/src/lib/lvm/libmeta/Makefile.com

2

```
60     meta_mem.o \
61     meta_metad.o \
62     meta_metad_subr.o \
63     meta_med.o \
64     meta_mh.o \
65     meta_mirror.o \
66     meta_mirror_resync.o \
67     meta_mn_comm.o \
68     meta_mn_changelog.o \
69     meta_mn_handlers.o \
70     meta_mn_msg_table.o \
71     meta_mn_subr.o \
72     meta_mount.o \
73     meta_name.o \
74     meta_nameinfo.o \
75     meta_namespace.o \
76     meta_notify.o \
77     meta_se_notify.o \
78     meta_patch.o \
79     meta_patch_root.o \
80     meta_print.o \
81     meta_raid.o \
82     meta_raid_resync.o \
83     meta_rename.o \
84     meta_repartition.o \
85     meta_replace.o \
86     meta_reset.o \
87     meta_resync.o \
88     meta_runtime.o \
89     meta_set.o \
90     meta_set_drv.o \
91     meta_set_hst.o \
92     meta_set_med.o \
93     meta_set_prv.o \
94     meta_set_tkr.o \
95     meta_setup.o \
96     meta_smf.o \
97     meta_stat.o \
98     meta_statconcise.o \
99     meta_sp.o \
100     meta_stripe.o \
101     meta_systemfile.o \
102     meta_tab.o \
103     meta_time.o \
104     meta_trans.o \
105     meta_userflags.o \
106     metarpopen.o \
107     metasplitname.o \
108     metagetroot.o \
109     sdssc_bind.o

111 SPC_OBJS= meta_check.o

113 CMN_SRCS = $(CMN_OBJS:%.o=$(COMMON)/%.c)
114 LOCAL_SRCS = $(LOCAL_OBJS:%.o=../common/%.c)
115 DERIVED_SRCS = $(DERIVED_OBJS:%.o=%.c)
116 SPC_SRCS = $(SPC_OBJS:%.o=../common/%.c)

118 OBJECTS64 = $(LOCAL_OBJS) $(DERIVED_OBJS) $(CMN_OBJS)
119 OBJECTS = $(OBJECTS64) $(SPC_OBJS)

121 include $(SRC)/lib/lvm/Makefile.lvm

123 MSGSRCS = $(LOCAL_SRCS) $(SPC_SRCS)
124 MSGFILES = $(MSGSRCS:%.c=%.i)
125 POFILE = libmeta.po
```

```

127 DCFILES = ../common/meta_print.po
128 DCFILE = libmeta.dc

130 CLOBBERFILES += $(POFILE) $(DCFILE)

132 # install this library in the root filesystem
133 include ../../../../Makefile.rootfs

135 LIBS = $(DYNLIB) $(LINTLIB)
136 SRCS = $(CMN_SRCS) $(LOCAL_SRCS) $(DERIVED_SRCS)
137 $(LINTLIB) := SRCS = $(SRCDIR)/$(LINTSRC)
138 lint := SRCS = $(CMN_SRCS) $(LOCAL_SRCS) $(SPC_SRCS)
139 CPPFLAGS += -I$(SRC)/lib/lvm/libmeta/common/hdrs
140 LDLIBS += -lnsl -lc -ladm -ldev -lgen -lefi -ldevinfo -lscf
141 CLEANFILES += $(DERIVED_SRCS) $(MSGFILES) $(DCFILES)

143 .KEEP_STATE:

145 BIG_TARGETS = $(OBJECTS64:%=pics/%)

147 $(BIG_TARGETS) := CPPFLAGS += -D_LARGEFILE_SOURCE=1 -D_FILE_OFFSET_BITS=64

149 $(LINTLIB) := CPPFLAGS += -D_LARGEFILE_SOURCE=1 -D_FILE_OFFSET_BITS=64

151 all: $(LIBS)

153 install debug: all $(ROOTLIBS) $(ROOTLINT) $(ROOTLINKS)

155 objs/%.o profs/%.o pics/%.o: $(COMMON)/%.c
156 $(COMPILE.c) -o $@ $<
157 $(POST_PROCESS_O)

159 mdiox_xdr.c: $(SRC)/uts/common/sys/lvm/mdiox.x
160 $(RPCGEN) $(RPCGENFLAGS) -c -i 100 $(SRC)/uts/common/sys/lvm/mdiox.x | \
161 /usr/xpg4/bin/awk '{sub(/uts\common\sys\lvm/, "head"); print $$0}' >
162 awk '{sub(/uts\common\sys\lvm/, "head"); print $$0}' >$@

163 meta_basic_xdr.c: $(SRC)/uts/common/sys/lvm/meta_basic.x
164 $(RPCGEN) $(RPCGENFLAGS) -c $(SRC)/uts/common/sys/lvm/meta_basic.x | \
165 /usr/xpg4/bin/awk '{sub(/uts\common\sys\lvm/, "head"); print $$0}' >
166 awk '{sub(/uts\common\sys\lvm/, "head"); print $$0}' >$@

167 metad_clnt.c: $(SRC)/head/metad.x
168 $(RPCGEN) $(RPCGENFLAGS) -l $(SRC)/head/metad.x -o $@

170 metad_xdr.c: $(SRC)/head/metad.x
171 $(RPCGEN) $(RPCGENFLAGS) -c $(SRC)/head/metad.x -o $@

173 metamed_clnt.c: $(SRC)/uts/common/sys/lvm/metamed.x
174 $(RPCGEN) $(RPCGENFLAGS) -l $(SRC)/uts/common/sys/lvm/metamed.x | \
175 /usr/xpg4/bin/awk '{sub(/uts\common\sys\lvm/, "head"); print $$0}' >
176 awk '{sub(/uts\common\sys\lvm/, "head"); print $$0}' >$@

177 metamed_xdr.c: $(SRC)/uts/common/sys/lvm/metamed.x
178 $(RPCGEN) $(RPCGENFLAGS) -c $(SRC)/uts/common/sys/lvm/metamed.x | \
179 /usr/xpg4/bin/awk '{sub(/uts\common\sys\lvm/, "head"); print $$0}' >
180 awk '{sub(/uts\common\sys\lvm/, "head"); print $$0}' >$@

181 metamhd_clnt.c: $(SRC)/head/metamhd.x
182 $(RPCGEN) $(RPCGENFLAGS) -l $(SRC)/head/metamhd.x -o $@

184 metamhd_xdr.c: $(SRC)/head/metamhd.x
185 $(RPCGEN) $(RPCGENFLAGS) -c $(SRC)/head/metamhd.x -o $@

187 mhd_xdr.c: $(SRC)/uts/common/sys/lvm/mhd_x.x

```

```

188 $(RPCGEN) $(RPCGENFLAGS) -c $(SRC)/uts/common/sys/lvm/mhd_x.x | \
189 /usr/xpg4/bin/awk '{sub(/uts\common\sys\lvm/, "head"); print $$0}' >
190 awk '{sub(/uts\common\sys\lvm/, "head"); print $$0}' >$@

191 mdmn_commd_xdr.c: $(SRC)/uts/common/sys/lvm/mdmn_commd.x
192 $(RPCGEN) -c $(SRC)/uts/common/sys/lvm/mdmn_commd.x -o $@

194 $(DCFILE) := XGETFLAGS = -c TRANSLATION_NOTE_LC_TIME -t

196 $(DCFILE): $(DCFILES)
197 $(CAT) $(DCFILES) > $(DCFILE)

199 $(POFILE): $(MSGFILES)
200 $(BUILDPO.msgfiles)

202 _msg: $(MSGDOMAINPOFILE)

204 _dc: $(DCMSGDOMAINPOFILE)

206 include $(SRC)/lib/lvm/Makefile.targ
207 include $(SRC)/Makefile.msg.targ

```

new/usr/src/lib/smb/rv/lib/smb/Makefile

1

1179 Sun May 26 18:23:32 2013

new/usr/src/lib/smb/rv/lib/smb/Makefile

XXX Remove awk(1)

```
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) 2007, 2010, Oracle and/or its affiliates. All rights reserved.
23 #
24 #
25 HDRS=  lib/smb.h
26
27 GENSRCS = common/smb_status.tbl.h
28 CLEANFILES += $(GENSRCS)
29
30 all install lint: $(GENSRCS)
31
32 include ../Makefile.smb/rv
33
34 $(GENSRCS): common/smb_status_gen.awk $(SRC)/uts/common/smb/ntstatus.h
35 /usr/xpg4/bin/awk -f common/smb_status_gen.awk \
36   awk -f common/smb_status_gen.awk \
37   $(SRC)/uts/common/smb/ntstatus.h > $@
```

new/usr/src/man/man1/Makefile

1

```
*****  
13370 Sun May 26 18:23:32 2013  
new/usr/src/man/man1/Makefile  
XXX Remove awk(1)  
*****  
_____unchanged_portion_omitted_
```

```

*****
12816 Sun May 26 18:23:33 2013
new/usr/src/man/man1/cputrack.1
XXX Remove awk(1)
*****
1 \" te
2.\" Copyright (c) 2004, Sun Microsystems, Inc.
3.\" The contents of this file are subject to the terms of the Common Development
4.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
5.\" When distributing Covered Code, include this CDDL HEADER in each file and in
6.TH CPUTRACK 1 "Apr 19, 2004"
7.SH NAME
8 cputrack \- monitor process and LWP behavior using CPU performance counters
9.SH SYNOPSIS
10.LP
11.nf
12 \fBcputrack\fR \fB-c\fR \fIeventspec\fR [\fB-c\fR \fIeventspec\fR]... [\fB-efntv\fR
13 [\fB-N\fR \fIcount\fR] [\fB-o\fR \fIpathname\fR] [\fB-T\fR \fIinterval\fR]
14 .fi

16.LP
17.nf
18 \fBcputrack\fR \fB-c\fR \fIeventspec\fR [\fB-c\fR \fIeventspec\fR]... \fB-p\fR \
19 [\fB-N\fR \fIcount\fR] [\fB-o\fR \fIpathname\fR] [\fB-T\fR \fIinterval\fR]
20 .fi

22.LP
23.nf
24 \fBcputrack\fR \fB-h\fR
25 .fi

27.SH DESCRIPTION
28.sp
29.LP
30 The \fBcputrack\fR utility allows \fBCPU\fR performance counters to be used to
31 monitor the behavior of a process or family of processes running on the system.
32 If \fIinterval\fR is specified with the \fB-T\fR option, \fBcputrack\fR samples
33 activity every \fIinterval\fR seconds, repeating forever. If a \fIcount\fR is
34 specified with the \fB-N\fR option, the statistics are repeated \fIcount\fR
35 times for each process tracked. If neither are specified, an interval of one
36 second is used. If \fIcommand\fR and optional \fIargs\fR are specified,
37 \fBcputrack\fR runs the command with the arguments given while monitoring the
38 specified \fBCPU\fR performance events. Alternatively, the process \fBID\fR of
39 an existing process can be specified using the \fB-p\fR option.
40 .sp
41.LP
42 Because \fBcputrack\fR is an unprivileged program, it is subject to the same
43 restrictions that apply to \fBtruss\fR(1). For example, \fBsetuid\fR(2)
44 executables cannot be tracked.
45.SH OPTIONS
46.sp
47.LP
48 The following options are supported:
49 .sp
50 .ne 2
51 .na
52 \fB\fB-c\fR \fIeventspec\fR\fR
53 .ad
54 .RS 16n
55 Specifies a set of events for the \fBCPU\fR performance counters to monitor.
56 The syntax of these event specifications is:
57 .sp
58 .in +2
59 .nf
60 [picn=]\fIeventn\fR[,attr[\fIn\fR][=\fIval\fR]][, [picn=]\fIeventn\fR
61 [\fIattr[n][=\fIval\fR]],...]
```

```

62 .fi
63 .in -2
64 .sp

66 You can use the \fB-h\fR option to obtain a list of available events and
67 attributes. This causes generation of the usage message. You can omit an
68 explicit counter assignment, in which case \fBcpustat\fR attempts to choose a
69 capable counter automatically.
70 .sp
71 Attribute values can be expressed in hexadecimal, octal, or decimal notation,
72 in a format suitable for \fBstrtoll\fR(3C). An attribute present in the event
73 specification without an explicit value receives a default value of \fB1\fR. An
74 attribute without a corresponding counter number is applied to all counters in
75 the specification.
76 .sp
77 The semantics of these event specifications can be determined by reading the
78 \fBCPU\fR manufacturer's documentation for the events.
79 .sp
80 Multiple \fB-c\fR options can be specified, in which case \fBcputrack\fR cycles
81 between the different event settings on each sample.
82 .RE

84 .sp
85 .ne 2
86 .na
87 \fB\fB-D\fR\fR
88 .ad
89 .RS 16n
90 Enables debug mode.
91 .RE

93 .sp
94 .ne 2
95 .na
96 \fB\fB-e\fR\fR
97 .ad
98 .RS 16n
99 Follows all \fBexec\fR(2), or \fBexecve\fR(2) system calls.
100 .RE

102 .sp
103 .ne 2
104 .na
105 \fB\fB-f\fR\fR
106 .ad
107 .RS 16n
108 Follows all children created by \fBfork\fR(2), \fBfork1\fR(2), or
109 \fBvfork\fR(2) system calls.
110 .RE

112 .sp
113 .ne 2
114 .na
115 \fB\fB-h\fR\fR
116 .ad
117 .RS 16n
118 Prints an extended help message on how to use the utility, how to program the
119 processor-dependent counters, and where to look for more detailed information.
120 .RE

122 .sp
123 .ne 2
124 .na
125 \fB\fB-n\fR\fR
126 .ad
127 .RS 16n
```


128 Omits all header output (useful if \fBcputrack\fR is the beginning of a
 129 pipeline).
 130 .RE

132 .sp
 133 .ne 2
 134 .na
 135 \fB\fB-N\fR \fBIcount\fR\fR
 136 .ad
 137 .RS 16n
 138 Specifies the maximum number of \fBCPU\fR performance counter samples to take
 139 before exiting.
 140 .RE

142 .sp
 143 .ne 2
 144 .na
 145 \fB\fB-o\fR \fIoutfile\fR\fR
 146 .ad
 147 .RS 16n
 148 Specifies file to be used for the \fBcputrack\fR output.
 149 .RE

151 .sp
 152 .ne 2
 153 .na
 154 \fB\fB-p\fR \fIpid\fR\fR
 155 .ad
 156 .RS 16n
 157 Interprets the argument as the process \fBID\fR of an existing process to which
 158 process counter context should be attached and monitored.
 159 .RE

161 .sp
 162 .ne 2
 163 .na
 164 \fB\fB-t\fR\fR
 165 .ad
 166 .RS 16n
 167 Prints an additional column of processor cycle counts, if available on the
 168 current architecture.
 169 .RE

171 .sp
 172 .ne 2
 173 .na
 174 \fB\fB-T\fR \fIinterval\fR\fR
 175 .ad
 176 .RS 16n
 177 Specifies the interval between \fBCPU\fR performance counter samples in
 178 seconds. Very small intervals may cause some samples to be skipped. See
 179 WARNINGS.
 180 .RE

182 .sp
 183 .ne 2
 184 .na
 185 \fB\fB-v\fR\fR
 186 .ad
 187 .RS 16n
 188 Enables more verbose output.
 189 .RE

191 .SH USAGE
 192 .sp
 193 .LP

194 The operating system enforces certain restrictions on the tracing of processes.
 195 In particular, a command whose object file cannot be read by a user cannot be
 196 tracked by that user; set-uid and set-gid commands can only be tracked by a
 197 privileged user. Unless it is run by a privileged user, \fBcputrack\fR loses
 198 control of any process that performs an \fBExec()\fR of a set-id or unreadable
 199 object file. Such processes continue normally, though independently of
 200 \fBcputrack\fR, from the point of the \fBExec()\fR.
 201 .sp
 202 .LP
 203 The system may run out of per-user process slots when the \fB-f\fR option is
 204 used, since \fBcputrack\fR runs one controlling process for each process being
 205 tracked.
 206 .sp
 207 .LP
 208 The times printed by \fBcputrack\fR correspond to the wallclock time when the
 209 hardware counters were actually sample. The time is derived from the same
 210 timebase as \fBgethrtime\fR(3C).
 211 .sp
 212 .LP
 213 The \fBcputrack\fR utility attaches performance counter context to each process
 214 that it examines. The presence of this context allows the performance counters
 215 to be multiplexed between different processes on the system, but it cannot be
 216 used at the same time as the \fBcpustat\fR(1M) utility.
 217 .sp
 218 .LP
 219 Once an instance of the \fBcpustat\fR utility is running, further attempts to
 220 run \fBcputrack\fR will fail until all instances of \fBcpustat\fR terminate.
 221 .sp
 222 .LP
 223 Sometimes \fBcputrack\fR provides sufficient flexibility and prints sufficient
 224 statistics to make adding the observation code to an application unnecessary.
 225 However, more control is occasionally desired. Because the same performance
 226 counter context is used by both the application itself and by the agent LWP
 227 injected into the application by \fBcputrack\fR, it is possible for an
 228 application to interact with the counter context to achieve some interesting
 229 capabilities. See \fBcpc_enable\fR(3CPC).
 230 .sp
 231 .LP
 232 The processor cycle counts enabled by the \fB-t\fR option always apply to both
 233 user and system modes, regardless of the settings applied to the performance
 234 counter registers.
 235 .sp
 236 .LP
 237 The output of \fBcputrack\fR is designed to be readily parseable by
 238 **\fBawk\fR(1) and \fBperl\fR(1), thereby allowing performance tools to be**
 238 *\fBnawk\fR(1) and \fBperl\fR(1), thereby allowing performance tools to be*
 239 composed by embedding \fBcputrack\fR in scripts. Alternatively, tools may be
 240 constructed directly using the same \fBBAPI\fRs that \fBcputrack\fR is built
 241 upon, using the facilities of \fBlibcpc\fR(3LIB) and \fBlibpcctx\fR(3LIB). See
 242 \fBcpc\fR(3CPC).
 243 .sp
 244 .LP
 245 Although \fBcputrack\fR uses performance counter context to maintain separate
 246 performance counter values for each LWP, some of the events that can be counted
 247 will inevitably be impacted by other activities occurring on the system,
 248 particularly for limited resources that are shared between processes (for
 249 example, cache miss rates). For such events, it may also be interesting to
 250 observe overall system behavior with \fBcpustat\fR(1M).
 251 .sp
 252 .LP
 253 For the \fB-T\fR \fIinterval\fR option, if \fIinterval\fR is specified as zero,
 254 no periodic sampling is performed. The performance counters are only sampled
 255 when the process creates or destroys an \fBBLWP\fR, or it invokes \fBfork\fR(2),
 256 \fBExec\fR(2), or \fBExit\fR(2).
 257 .SH EXAMPLES
 258 .SS "SPARC"

```

259 .LP
260 \fBExample 1 \fRUsing Performance Counters to Count Clock Cycles
261 .sp
262 .LP
263 In this example, the utility is being used on a machine containing an
264 UltraSPARC-III+ processor. The counters are set to count processor clock cycles
265 and instructions dispatched in user mode while running the \fBsleep\fR(1)
266 command.

```

```

268 .sp
269 .in +2
270 .nf
271 example% \fBcputrack -c pic0=Cycle_cnt,pic1=Instr_cnt sleep 10\fR

```

274	time	lwp	event	pic0	pic1
275	1.007	1	tick	765308	219233
276	2.007	1	tick	0	0
277	4.017	1	tick	0	0
278	6.007	1	tick	0	0
279	8.007	1	tick	0	0
280	10.007	1	tick	0	0
281	10.017	1	exit	844703	228058

```

283 .fi
284 .in -2
285 .sp

```

```

287 .LP
288 \fBExample 2 \fRCounting External Cache References and Misses
289 .sp
290 .LP
291 This example shows more verbose output while following the \fBfork()\fR and
292 \fBexec()\fR of a simple shell script on an UltraSPARC machine. The counters
293 are measuring the number of external cache references and external cache
294 misses. Notice that the explicit \fBpic0\fR and \fBpic1\fR names can be omitted
295 where there are no ambiguities.

```

```

297 .sp
298 .in +2
299 .nf
300 example% \fBcputrack -fev -c EC_ref,EC_hit /bin/ulimit -c\fR

```

303	time	pid	lwp	event	pic0	pic1
304	0.007	101142	1	init_lwp	805286	20023
305	0.023	101142	1	fork		# 101143
306	0.026	101143	1	init_lwp	1015382	24461
307	0.029	101143	1	fini_lwp	1025546	25074
308	0.029	101143	1	exec	1025546	25074
309	0.000	101143	1	exec		\e
310						# '/usr/bin/sh /usr/bin/basename\e
311						/bin/ulimit'
312	0.039	101143	1	init_lwp	1025546	25074
313	0.050	101143	1	fini_lwp	1140482	27806
314	0.050	101143	1	exec	1140482	27806
315	0.000	101143	1	exec		# '/usr/bin/expr \e
316						//bin/ulimit : \(. *[^\)]\)/*\$: .*\/\(..*\) : \(. *\)\$ //bin/ulimi'
317	0.059	101143	1	init_lwp	1140482	27806
318	0.075	101143	1	fini_lwp	1237647	30207
319	0.075	101143	1	exit	1237647	30207
320						unlimited
321	0.081	101142	1	fini_lwp	953383	23814
322	0.081	101142	1	exit	953383	23814
323						.fi
324						.in -2

```

325 .sp
327 .SS "x86"
328 .LP
329 \fBExample 3 \fRCounting Instructions
330 .sp
331 .LP
332 This example shows how many instructions were executed in the application and
333 in the kernel to print the date on a Pentium III machine:

```

```

335 .sp
336 .in +2
337 .nf
338 example% \fBcputrack -c inst_retired,inst_retired,nouser1,sys1 date\fR

```

341	time	lwp	event	pic0	pic1
342	Fri Aug 20 20:03:08 PDT 1999				
343	0.072	1	exit	246725	339666
344					.fi
345					.in -2
346					.sp

```

348 .LP
349 \fBExample 4 \fRCounting TLB Hits
350 .sp
351 .LP
352 This example shows how to use processor-specific attributes to count TLB hits
353 on a Pentium 4 machine:

```

```

355 .sp
356 .in +2
357 .nf
358 example% \fBcputrack -c ITLB_reference,emask=1 date\fR

```

361	time	lwp	event	pic0
362	Fri Aug 20 20:03:08 PDT 1999			
363	0.072	1	exit	246725
364				.fi
365				.in -2
366				.sp

```

368 .SH WARNINGS
369 .sp
370 .LP
371 By running any instance of the \fBcpustat\fR(1M) utility, all existing
372 performance counter context is forcibly invalidated across the machine. This
373 may in turn cause all invocations of the \fBcputrack\fR command to exit
374 prematurely with unspecified errors.
375 .sp
376 .LP
377 If \fBcpustat\fR is invoked on a system that has \fBFCPU\fR performance counters
378 which are not supported by Solaris, the following message appears:
379 .sp
380 .in +2
381 .nf
382 cputrack: cannot access performance counters - Operation not applicable
383 .fi
384 .in -2
385 .sp

```

```

387 .sp
388 .LP
389 This error message implies that \fBcpc_open()\fR has failed and is documented
390 in \fBcpc_open\fR(3CPC). Review this documentation for more information about

```

```
391 the problem and possible solutions.
392 .sp
393 .LP
394 If a short interval is requested, \fBcputrack\fR may not be able to keep up
395 with the desired sample rate. In this case, some samples may be dropped.
396 .SH ATTRIBUTES
397 .sp
398 .LP
399 See \fBattributes\fR(5) for descriptions of the following attributes:
400 .sp
402 .sp
403 .TS
404 box;
405 c | c
406 l | l .
407 ATTRIBUTE TYPE ATTRIBUTE VALUE
408 Interface Stability Evolving
409 .TE
411 .SH SEE ALSO
412 .sp
413 .LP
414 \fBawk\fR(1), \fBperl\fR(1), \fBproc\fR(1), \fBtruss\fR(1), \fBprstat\fR(1M),
414 \fBbnawk\fR(1), \fBper1\fR(1), \fBproc\fR(1), \fBtruss\fR(1), \fBprstat\fR(1M),
415 \fBcpustat\fR(1M), \fBexec\fR(2), \fBexit\fR(2), \fBfork\fR(2),
416 \fBsetuid\fR(2), \fBvfork\fR(2), \fBgethrtime\fR(3C), \fBstrtoll\fR(3C),
417 \fBcpc\fR(3CPC), \fBcpc_bind_pctx\fR(3CPC), \fBcpc_enable\fR(3CPC),
418 \fBcpc_open\fR(3CPC), \fBlibcpc\fR(3LIB), \fBlibpctx\fR(3LIB), \fBproc\fR(4),
419 \fBattributes\fR(5)
```

57515 Sun May 26 18:23:33 2013

new/usr/src/man/man1/nawk.1

XXX Remove nawk(1)

```

1 \" te
2.\" Copyright 1989 AT&T
3.\" Copyright 1992, X/Open Company Limited All Rights Reserved
4.\" Portions Copyright (c) 2005, 2006 Sun Microsystems, Inc. All Rights Reserve
5.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for permission
6.\" http://www.opengroup.org/bookstore/.
7.\" The Institute of Electrical and Electronics Engineers and The Open Group, ha
8.\" This notice shall appear on any product containing this material.
9.\" The contents of this file are subject to the terms of the Common Development
10.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
11.\" When distributing Covered Code, include this CDDL HEADER in each file and in
12.TH NAWK 1 "May 24, 2006"
13.SH NAME
14 nawk \- pattern scanning and processing language
15.SH SYNOPSIS
16.LP
17.nf
18 \fB/usr/bin/nawk\fR [\fB-F\fR \fIERE\fR] [\fB-v\fR \fIassignment\fR] \fI\&'progr
19 [\fIargument\fR]...
20.fi

22.LP
23.nf
18 \fB/usr/xpg4/bin/awk\fR [\fB-F\fR \fIERE\fR] [\fB-v\fR \fIassignment\fR]... \fI\
19 [\fIargument\fR]...
20.fi

22.SH DESCRIPTION
23.sp
24.LP
25 The \fB/usr/xpg4/bin/awk\fR utility executes
31 The \fB/usr/bin/nawk\fR and \fB/usr/xpg4/bin/awk\fR utilities execute
26 \fIprogram\fRs written in the \fBnawk\fR programming language, which is
27 specialized for textual data manipulation. A \fBnawk\fR \fIprogram\fR is a
28 sequence of patterns and corresponding actions. The string specifying
29 \fIprogram\fR must be enclosed in single quotes (') to protect it from
30 interpretation by the shell. The sequence of pattern - action statements can be
31 specified in the command line as \fIprogram\fR or in one, or more, file(s)
32 specified by the \fB-f\fR \fIprogfile\fR option. When input is read that matches
33 a pattern, the action associated with the pattern is performed.
34.sp
35.LP
36 Input is interpreted as a sequence of records. By default, a record is a line,
37 but this can be changed by using the \fBRS\fR built-in variable. Each record of
38 input is matched to each pattern in the \fIprogram\fR. For each pattern
39 matched, the associated action is executed.
40.sp
41.LP
42 The \fBnawk\fR utility interprets each input record as a sequence of fields
43 where, by default, a field is a string of non-blank characters. This default
44 white-space field delimiter (blanks and/or tabs) can be changed by using the
45 \fBFS\fR built-in variable or the \fB-F\fR \fIERE\fR option. The \fBnawk\fR
46 utility denotes the first field in a record \fB$1\fR, the second \fB$2\fR, and
47 so forth. The symbol \fB$0\fR refers to the entire record; setting any other
48 field causes the reevaluation of \fB$0\fR. Assigning to \fB$0\fR resets the
49 values of all fields and the \fBNF\fR built-in variable.
50.SH OPTIONS
51.sp
52.LP
53 The following options are supported:
54.sp

```

```

55.ne 2
56.na
57 \fB\F\fR \fIERE\fR \fR
58.ad
59.RS 17n
60 Define the input field separator to be the extended regular expression
61 \fIERE\fR, before any input is read (can be a character).
62.RE

64.sp
65.ne 2
66.na
67 \fB\F\fR \fIprogfile\fR \fR
68.ad
69.RS 17n
70 Specifies the pathname of the file \fIprogfile\fR containing a \fBnawk\fR
71 program. If multiple instances of this option are specified, the concatenation
72 of the files specified as \fIprogfile\fR in the order specified is the
73 \fBnawk\fR program. The \fBnawk\fR program can alternatively be specified in
74 the command line as a single argument.
75.RE

77.sp
78.ne 2
79.na
80 \fB\F-v\fR \fIassignment\fR \fR
81.ad
82.RS 17n
83 The \fIassignment\fR argument must be in the same form as an \fIassignment\fR
84 operand. The assignment is of the form \fIvar=value\fR, where \fIvar\fR is the
85 name of one of the variables described below. The specified assignment occurs
86 before executing the \fBnawk\fR program, including the actions associated with
87 \fBBEGIN\fR patterns (if any). Multiple occurrences of this option can be
88 specified.
89.RE

91.SH OPERANDS
92.sp
93.LP
94 The following operands are supported:
95.sp
96.ne 2
97.na
98 \fB\F\fR \fIprogram\fR \fR
99.ad
100.RS 12n
101 If no \fB-f\fR option is specified, the first operand to \fBnawk\fR is the text
102 of the \fBnawk\fR program. The application supplies the \fIprogram\fR operand
103 as a single argument to \fBnawk.\fR If the text does not end in a newline
104 character, \fBnawk\fR interprets the text as if it did.
105.RE

107.sp
108.ne 2
109.na
110 \fB\F\fR \fIargument\fR \fR
111.ad
112.RS 12n
113 Either of the following two types of \fIargument\fR can be intermixed:
114.sp
115.ne 2
116.na
117 \fB\F\fR \fIfile\fR \fR
118.ad
119.RS 14n
120 A pathname of a file that contains the input to be read, which is matched

```

121 against the set of patterns in the program. If no `\fifile\fR` operands are
 122 specified, or if a `\fifile\fR` operand is `\fB\(\mi\fR`, the standard input is
 123 used.
 124 .RE

126 .sp
 127 .ne 2
 128 .na
 129 `\fB\fIassignment\fR\fR`
 130 .ad
 131 .RS 14n
 132 An operand that begins with an underscore or alphabetic character from the
 133 portable character set, followed by a sequence of underscores, digits and
 134 alphabets from the portable character set, followed by the `\fB=\fR` character
 135 specifies a variable assignment rather than a pathname. The characters before
 136 the `\fB=\fR` represent the name of a `\fBnawk\fR` variable. If that name is a
 137 `\fBnawk\fR` reserved word, the behavior is undefined. The characters following
 138 the equal sign is interpreted as if they appeared in the `\fBnawk\fR` program
 139 preceded and followed by a double-quote (`\fB"\fR`) character, as a `\fBSTRING\fR`
 140 token, except that if the last character is an unescaped backslash, it is
 141 interpreted as a literal backslash rather than as the first character of the
 142 sequence `\fB\efR&..`. The variable is assigned the value of that `\fBSTRING\fR`
 143 token. If the value is considered a `\fInumeric\fRstring\fI,\fR` the variable is
 144 assigned its numeric value. Each such variable assignment is performed just
 145 before the processing of the following `\fifile\fR`, if any. Thus, an assignment
 146 before the first `\fBfile\fR` argument is executed after the `\fBBEGIN\fR` actions
 147 (if any), while an assignment after the last `\fifile\fR` argument is executed
 148 before the `\fBEND\fR` actions (if any). If there are no `\fifile\fR` arguments,
 149 assignments are executed before processing the standard input.
 150 .RE

152 .RE

154 .SH INPUT FILES
 155 .sp
 156 .LP
 157 Input files to the `\fBnawk\fR` program from any of the following sources:
 158 .RS +4
 159 .TP
 160 .ie t \(\bu
 161 .el o
 162 any `\fifile\fR` operands or their equivalents, achieved by modifying the
 163 `\fBnawk\fR` variables `\fBARGV\fR` and `\fBARGC\fR`
 164 .RE
 165 .RS +4
 166 .TP
 167 .ie t \(\bu
 168 .el o
 169 standard input in the absence of any `\fifile\fR` operands
 170 .RE
 171 .RS +4
 172 .TP
 173 .ie t \(\bu
 174 .el o
 175 arguments to the `\fBgetline\fR` function
 176 .RE
 177 .sp
 178 .LP
 179 must be text files. Whether the variable `\fBRS\fR` is set to a value other than
 180 a newline character or not, for these files, implementations support records
 181 terminated with the specified separator up to `\fB{LINE_MAX}\fR` bytes and can
 182 support longer records.
 183 .sp
 184 .LP
 185 If `\fB-\fR\fBf\fR` `\fIprogfile\fR` is specified, the files named by each of the
 186 `\fIprogfile\fR` option-arguments must be text files containing an `\fBnawk\fR`

187 program.
 188 .sp
 189 .LP
 190 The standard input are used only if no `\fifile\fR` operands are specified, or if
 191 a `\fifile\fR` operand is `\fB\(\mi\fR&.`
 192 .SH EXTENDED DESCRIPTION
 193 .sp
 194 .LP
 195 A `\fBnawk\fR` program is composed of pairs of the form:
 196 .sp
 197 .in +2
 198 .nf
 199 pattern { `\fIaction\fR` }
 200 .fi
 201 .in -2

203 .sp
 204 .LP
 205 Either the pattern or the action (including the enclosing brace characters) can
 206 be omitted. Pattern-action statements are separated by a semicolon or by a
 207 newline.
 208 .sp
 209 .LP
 210 A missing pattern matches any record of input, and a missing action is
 211 equivalent to an action that writes the matched record of input to standard
 212 output.
 213 .sp
 214 .LP
 215 Execution of the `\fBnawk\fR` program starts by first executing the actions
 216 associated with all `\fBBEGIN\fR` patterns in the order they occur in the
 217 program. Then each `\fifile\fR` operand (or standard input if no files were
 218 specified) is processed by reading data from the file until a record separator
 219 is seen (a newline character by default), splitting the current record into
 220 fields using the current value of `\fBFS\fR`, evaluating each pattern in the
 221 program in the order of occurrence, and executing the action associated with
 222 each pattern that matches the current record. The action for a matching pattern
 223 is executed before evaluating subsequent patterns. Last, the actions associated
 224 with all `\fBEND\fR` patterns is executed in the order they occur in the program.
 225 .SS "Expressions in nawk"
 226 .sp
 227 .LP
 228 Expressions describe computations used in `\fIpatterns\fR` and `\fIactions\fR`. In
 229 the following table, valid expression operations are given in groups from
 230 highest precedence first to lowest precedence last, with equal-precedence
 231 operators grouped between horizontal lines. In expression evaluation, where the
 232 grammar is formally ambiguous, higher precedence operators are evaluated before
 233 lower precedence operators. In this table `\fIexpr,\fR` `\fIexpr1,\fR`
 234 `\fIexpr2,\fR` and `\fIexpr3\fR` represent any expression, while `\fIvalue\fR`
 235 represents any entity that can be assigned to (that is, on the left side of an
 236 assignment operator).
 237 .sp

239 .sp
 240 .TS
 241 c c c c
 242 l l l l .
 243 `\fBSyntax\fR` `\fBName\fR` `\fBType of Result\fR` `\fBAssociativity\fR`
 244 -
 245 (`\fIexpr\fR`) Grouping type of `\fIexpr\fR` n/a
 246 -
 247 \$`\fIexpr\fR` Field reference string n/a
 248 -
 249 ++ `\fIvalue\fR` Pre-increment numeric n/a
 250 `\(\mi\(\mi\fIvalue\fR` Pre-decrement numeric n/a
 251 `\fIvalue\fR` ++ Post-increment numeric n/a
 252 `\fIvalue\fR` `\(\mi\(\mi` Post-decrement numeric n/a

```

253 _
254 \fIexpr\fR ^ \fIexpr\fR Exponentiation numeric right
255 _
256 ! \fIexpr\fR Logical not numeric n/a
257 + \fIexpr\fR Unary plus numeric n/a
258 ~(mi \fIexpr\fR Unary minus numeric n/a
259 _
260 \fIexpr\fR * \fIexpr\fR Multiplication numeric left
261 \fIexpr\fR / \fIexpr\fR Division numeric left
262 \fIexpr\fR % \fIexpr\fR Modulus numeric left
263 _
264 \fIexpr\fR + \fIexpr\fR Addition numeric left
265 \fIexpr\fR -(mi \fIexpr\fR Subtraction numeric left
266 _
267 \fIexpr\fR \fIexpr\fR String concatenation string left
268 _
269 \fIexpr\fR < \fIexpr\fR Less than numeric none
270 \fIexpr\fR <= \fIexpr\fR Less than or equal to numeric none
271 \fIexpr\fR != \fIexpr\fR Not equal to numeric none
272 \fIexpr\fR == \fIexpr\fR Equal to numeric none
273 \fIexpr\fR > \fIexpr\fR Greater than numeric none
274 \fIexpr\fR >= \fIexpr\fR Greater than or equal to numeric none
275 _
276 \fIexpr\fR ~ \fIexpr\fR ERE match numeric none
277 \fIexpr\fR !~ \fIexpr\fR ERE non-match numeric none
278 _
279 \fIexpr\fR in array Array membership numeric left
280 ( \fIindex\fR ) in Multi-dimension array numeric left
281 \fIarray\fR membership
282 _
283 \fBexpr\fR && \fIexpr\fR Logical AND numeric left
284 _
285 \fBexpr\fR || \fIexpr\fR Logical OR numeric left
286 _
287 \fIexpr1\fR ? \fIexpr2\fR Conditional expression type of selected
288 : \fIexpr3\fR \fIexpr2\fR or \fIexpr3\fR
289 _
290 \fIilvalue\fR ^= \fIexpr\fR Exponentiation numeric right
291 assignment
292 \fIilvalue\fR %= \fIexpr\fR Modulus assignment numeric right
293 \fIilvalue\fR *= \fIexpr\fR Multiplication numeric right
294 assignment
295 \fIilvalue\fR /= \fIexpr\fR Division assignment numeric right
296 \fIilvalue\fR += \fIexpr\fR Addition assignment numeric right
297 \fIilvalue\fR -(mi= \fIexpr\fR Subtraction assignment numeric right
298 \fIilvalue\fR = \fIexpr\fR Assignment type of \fIexpr\fR right
299 .TE

301 .sp
302 .LP
303 Each expression has either a string value, a numeric value or both. Except as
304 stated for specific contexts, the value of an expression is implicitly
305 converted to the type needed for the context in which it is used. A string
306 value is converted to a numeric value by the equivalent of the following calls:
307 .sp
308 .in +2
309 .nf
310 setlocale(LC_NUMERIC, "");
311 \fInumeric_value\fR = atof(\fIstring_value\fR);
312 .fi
313 .in -2

315 .sp
316 .LP
317 A numeric value that is exactly equal to the value of an integer is converted
318 to a string by the equivalent of a call to the \fBsprintf\fR function with the

```

```

319 string \fB%d\fR as the \fBfmt\fR argument and the numeric value being converted
320 as the first and only \fIexpr\fR argument. Any other numeric value is
321 converted to a string by the equivalent of a call to the \fBsprintf\fR function
322 with the value of the variable \fBCONVFMT\fR as the \fBfmt\fR argument and the
323 numeric value being converted as the first and only \fIexpr\fR argument.
324 .sp
325 .LP
326 A string value is considered to be a \fInumeric string\fR in the following
327 case:
328 .RS +4
329 .TP
330 1.
331 Any leading and trailing blank characters is ignored.
332 .RE
333 .RS +4
334 .TP
335 2.
336 If the first unignored character is a \fB+\fR or \fB(mi\fR, it is ignored.
337 .RE
338 .RS +4
339 .TP
340 3.
341 If the remaining unignored characters would be lexically recognized as a
342 \fBNUMBER\fR token, the string is considered a \fInumeric string\fR.
343 .RE
344 .sp
345 .LP
346 If a \fB(mi\fR character is ignored in the above steps, the numeric value of
347 the \fInumeric string\fR is the negation of the numeric value of the recognized
348 \fBNUMBER\fR token. Otherwise the numeric value of the \fInumeric string\fR is
349 the numeric value of the recognized \fBNUMBER\fR token. Whether or not a string
350 is a \fInumeric string\fR is relevant only in contexts where that term is used
351 in this section.
352 .sp
353 .LP
354 When an expression is used in a Boolean context, if it has a numeric value, a
355 value of zero is treated as false and any other value is treated as true.
356 Otherwise, a string value of the null string is treated as false and any other
357 value is treated as true. A Boolean context is one of the following:
358 .RS +4
359 .TP
360 .ie t \(\bu
361 .el o
362 the first subexpression of a conditional expression.
363 .RE
364 .RS +4
365 .TP
366 .ie t \(\bu
367 .el o
368 an expression operated on by logical NOT, logical \fBAND\fR or logical OR.
369 .RE
370 .RS +4
371 .TP
372 .ie t \(\bu
373 .el o
374 the second expression of a \fBfor\fR statement.
375 .RE
376 .RS +4
377 .TP
378 .ie t \(\bu
379 .el o
380 the expression of an \fBif\fR statement.
381 .RE
382 .RS +4
383 .TP
384 .ie t \(\bu

```

```

385 .el o
386 the expression of the \fBwhile\fR clause in either a \fBwhile\fR or \fBdo\fR
387 \fB\&.\|.\|.\fR \fBwhile\fR statement.
388 .RE
389 .RS +4
390 .TP
391 .ie t \(\bu
392 .el o
393 an expression used as a pattern (as in Overall Program Structure).
394 .RE
395 .sp
396 .LP
397 The \fBnawk\fR language supplies arrays that are used for storing numbers or
398 strings. Arrays need not be declared. They are initially empty, and their sizes
399 changes dynamically. The subscripts, or element identifiers, are strings,
400 providing a type of associative array capability. An array name followed by a
401 subscript within square brackets can be used as an \fIvalue\fR and as an
402 expression, as described in the grammar. Unsubscripted array names are used in
403 only the following contexts:
404 .RS +4
405 .TP
406 .ie t \(\bu
407 .el o
408 a parameter in a function definition or function call.
409 .RE
410 .RS +4
411 .TP
412 .ie t \(\bu
413 .el o
414 the \fBNAME\fR token following any use of the keyword \fBin\fR.
415 .RE
416 .sp
417 .LP
418 A valid array \fIindex\fR consists of one or more comma-separated expressions,
419 similar to the way in which multi-dimensional arrays are indexed in some
420 programming languages. Because \fBnawk\fR arrays are really one-dimensional,
421 such a comma-separated list is converted to a single string by concatenating
422 the string values of the separate expressions, each separated from the other by
423 the value of the \fBSUBSEP\fR variable.
424 .sp
425 .LP
426 Thus, the following two index operations are equivalent:
427 .sp
428 .in +2
429 .nf
430 var[expr1, expr2, ... exprn]
431 var[expr1 SUBSEP expr2 SUBSEP ... SUBSEP exprn]
432 .fi
433 .in -2

435 .sp
436 .LP
437 A multi-dimensioned \fIindex\fR used with the \fBin\fR operator must be put in
438 parentheses. The \fBin\fR operator, which tests for the existence of a
439 particular array element, does not create the element if it does not exist.
440 Any other reference to a non-existent array element automatically creates it.
441 .SS "Variables and Special Variables"
442 .sp
443 .LP
444 Variables can be used in an \fBnawk\fR program by referencing them. With the
445 exception of function parameters, they are not explicitly declared.
446 Uninitialized scalar variables and array elements have both a numeric value of
447 zero and a string value of the empty string.
448 .sp
449 .LP
450 Field variables are designated by a \fB$\fR followed by a number or numerical

```

```

451 expression. The effect of the field number \fIexpression\fR evaluating to
452 anything other than a non-negative integer is unspecified. Uninitialized
453 variables or string values need not be converted to numeric values in this
454 context. New field variables are created by assigning a value to them.
455 References to non-existent fields (that is, fields after \fB$NF\fR) produce the
456 null string. However, assigning to a non-existent field (for example,
457 \fB$(NF+2) = 5\fR) increases the value of \fBNF\fR, create any intervening
458 fields with the null string as their values and cause the value of \fB$0\fR to
459 be recomputed, with the fields being separated by the value of \fB$OFS\fR. Each
460 field variable has a string value when created. If the string, with any
461 occurrence of the decimal-point character from the current locale changed to a
462 period character, is considered a \fInumeric string\fR (see \fBExpressions in
463 nawk\fR above), the field variable also has the numeric value of the \fInumeric
464 string\fR.
465 .SS "/usr/bin/nawk, /usr/xpg4/bin/awk"
466 .sp
467 .LP
468 \fBnawk\fR sets the following special variables that are supported by both
469 \fB/usr/bin/nawk\fR and \fB/usr/xpg4/bin/awk\fR:
470 .sp
471 .ne 2
472 .na
473 \fBFBARGC\fR
474 .ad
475 .RS 12n
476 The number of elements in the \fBARGV\fR array.
477 .RE

479 .sp
480 .ne 2
481 .na
482 \fBFBARGV\fR
483 .ad
484 .RS 12n
485 An array of command line arguments, excluding options and the \fIprogram\fR
486 argument, numbered from zero to \fBARGC\fR\(\mil.
487 .sp
488 The arguments in \fBARGV\fR can be modified or added to; \fBARGC\fR can be
489 altered. As each input file ends, \fBnawk\fR treats the next non-null element
490 of \fBARGV\fR, up to the current value of \fBARGC\fR\(\mil, inclusive, as the
491 name of the next input file. Setting an element of \fBARGV\fR to null means
492 that it is not treated as an input file. The name \fB(mi\fR indicates the
493 standard input. If an argument matches the format of an \fIassignment\fR
494 operand, this argument is treated as an assignment rather than a \fIfile\fR
495 argument.
496 .RE

498 .sp
499 .ne 2
500 .na
501 \fBFBENVIRON\fR
502 .ad
503 .RS 12n
504 The variable \fBENVIRON\fR is an array representing the value of the
505 environment. The indices of the array are strings consisting of the names of
506 the environment variables, and the value of each array element is a string
507 consisting of the value of that variable. If the value of an environment
508 variable is considered a \fInumeric string\fR, the array element also has its
509 numeric value.
510 .sp
511 In all cases where \fBnawk\fR behavior is affected by environment variables
512 (including the environment of any commands that \fBnawk\fR executes via the
513 \fBsystem\fR function or via pipeline redirections with the \fBprint\fR
514 statement, the \fBprintf\fR statement, or the \fBgetline\fR function), the
515 environment used is the environment at the time \fBnawk\fR began executing.
516 .RE

```

```

518 .sp
519 .ne 2
520 .na
521 \fB\FBFILENAME\fR\fR
522 .ad
523 .RS 12n
524 A pathname of the current input file. Inside a \fBBEGIN\fR action the value is
525 undefined. Inside an \fBEND\fR action the value is the name of the last input
526 file processed.
527 .RE

529 .sp
530 .ne 2
531 .na
532 \fB\FBFNR\fR\fR
533 .ad
534 .RS 12n
535 The ordinal number of the current record in the current file. Inside a
536 \fBBEGIN\fR action the value is zero. Inside an \fBEND\fR action the value is
537 the number of the last record processed in the last file processed.
538 .RE

540 .sp
541 .ne 2
542 .na
543 \fB\FBFS\fR\fR
544 .ad
545 .RS 12n
546 Input field separator regular expression; a space character by default.
547 .RE

549 .sp
550 .ne 2
551 .na
552 \fB\FBNF\fR\fR
553 .ad
554 .RS 12n
555 The number of fields in the current record. Inside a \fBBEGIN\fR action, the
556 use of \fBNF\fR is undefined unless a \fBgetline\fR function without a
557 \fIvar\fR argument is executed previously. Inside an \fBEND\fR action, \fBNF\fR
558 retains the value it had for the last record read, unless a subsequent,
559 redirected, \fBgetline\fR function without a \fIvar\fR argument is performed
560 prior to entering the \fBEND\fR action.
561 .RE

563 .sp
564 .ne 2
565 .na
566 \fB\FBNR\fR\fR
567 .ad
568 .RS 12n
569 The ordinal number of the current record from the start of input. Inside a
570 \fBBEGIN\fR action the value is zero. Inside an \fBEND\fR action the value is
571 the number of the last record processed.
572 .RE

574 .sp
575 .ne 2
576 .na
577 \fB\FBOFMT\fR\fR
578 .ad
579 .RS 12n
580 The \fBprintf\fR format for converting numbers to strings in output statements
581 \fB"%g"\fR by default. The result of the conversion is unspecified if the
582 value of \fBOFMT\fR is not a floating-point format specification.

```

```

583 .RE

585 .sp
586 .ne 2
587 .na
588 \fB\FBOFS\fR\fR
589 .ad
590 .RS 12n
591 The \fBprint\fR statement output field separator; a space character by default.
592 .RE

594 .sp
595 .ne 2
596 .na
597 \fB\FBORS\fR\fR
598 .ad
599 .RS 12n
600 The \fBprint\fR output record separator; a newline character by default.
601 .RE

603 .sp
604 .ne 2
605 .na
606 \fB\FBLENGTH\fR\fR
607 .ad
608 .RS 12n
609 The length of the string matched by the \fBmatch\fR function.
610 .RE

612 .sp
613 .ne 2
614 .na
615 \fB\FBRS\fR\fR
616 .ad
617 .RS 12n
618 The first character of the string value of \fBRS\fR is the input record
619 separator; a newline character by default. If \fBRS\fR contains more than one
620 character, the results are unspecified. If \fBRS\fR is null, then records are
621 separated by sequences of one or more blank lines. Leading or trailing blank
622 lines do not produce empty records at the beginning or end of input, and the
623 field separator is always newline, no matter what the value of \fBFS\fR.
624 .RE

626 .sp
627 .ne 2
628 .na
629 \fB\FBRSTART\fR\fR
630 .ad
631 .RS 12n
632 The starting position of the string matched by the \fBmatch\fR function,
633 numbering from 1. This is always equivalent to the return value of the
634 \fBmatch\fR function.
635 .RE

637 .sp
638 .ne 2
639 .na
640 \fB\FBSUBSEP\fR\fR
641 .ad
642 .RS 12n
643 The subscript separator string for multi-dimensional arrays. The default value
644 is \fB"\e034"\fR.
645 .RE

647 .SS "/usr/xpg4/bin/awk"
648 .sp

```



```

649 .LP
650 The following variable is supported for \fB/usr/xpg4/bin/awk\fR only:
651 .sp
652 .ne 2
653 .na
654 \fB\fBCONVFMT\fR\fR
655 .ad
656 .RS 11n
657 The \fBprintf\fR format for converting numbers to strings (except for output
658 statements, where \fB%FMT\fR is used). The default is \fB%.6g\fR.
659 .RE

661 .SS "Regular Expressions"
662 .sp
663 .LP
664 The \fBnawk\fR utility makes use of the extended regular expression notation
665 (see \fBregex\fR(5)) except that it allows the use of C-language conventions to
666 escape special characters within the EREs, namely \fB\e\a\fR, \fB\ea\fR,
667 \fB\eb\fR, \fB\ef\fR, \fB\en\fR, \fB\er\fR, \fB\et\fR, \fB\ev\fR, and those
668 specified in the following table. These escape sequences are recognized both
669 inside and outside bracket expressions. Note that records need not be
670 separated by newline characters and string constants can contain newline
671 characters, so even the \fB\en\fR sequence is valid in \fBnawk\fR EREs. Using
672 a slash character within the regular expression requires escaping as shown in
673 the table below:
674 .sp

676 .sp
677 .TS
678 l l l
679 l l l .
680 \fB\Bescape Sequence\fR \fB\BDescription\fR \fB\BMeaning\fR
681 -
682 \fB\B\e"\fR Backslash quotation-mark Quotation-mark character
683 -
684 \fB\B\e/\fR Backslash slash Slash character
685 -
686 \fB\B\e\fR\fB\B\Iddd\fR T{
687 A backslash character followed by the longest sequence of one, two, or three oct
688 T{
689 The character encoded by the one-, two- or three-digit octal integer. Multi-byte
690 T}
691 -
692 \fB\B\e\fR\fB\B\Ic\fR T{
693 A backslash character followed by any character not described in this table or s
694 T} Undefined
695 .TE

697 .sp
698 .LP
699 A regular expression can be matched against a specific field or string by using
700 one of the two regular expression matching operators, \fB~\fR and \fB!~\fR.
701 These operators interpret their right-hand operand as a regular expression and
702 their left-hand operand as a string. If the regular expression matches the
703 string, the \fB~\fR expression evaluates to the value \fB1\fR, and the
704 \fB!~\fR expression evaluates to the value \fB0\fR. If the regular expression
705 does not match the string, the \fB~\fR expression evaluates to the value
706 \fB0\fR, and the \fB!~\fR expression evaluates to the value \fB1\fR. If the
707 right-hand operand is any expression other than the lexical token \fBBERE\fR,
708 the string value of the expression is interpreted as an extended regular
709 expression, including the escape conventions described above. Notice that these
710 same escape conventions also are applied in the determining the value of a
711 string literal (the lexical token \fBSTRING\fR), and is applied a second time
712 when a string literal is used in this context.
713 .sp
714 .LP

```

```

715 When an \fBERE\fR token appears as an expression in any context other than as
716 the right-hand of the \fB~\fR or \fB!~\fR operator or as one of the built-in
717 function arguments described below, the value of the resulting expression is
718 the equivalent of:
719 .sp
720 .in +2
721 .nf
722 $0 ~ /\fBIere\fR/
723 .fi
724 .in -2

726 .sp
727 .LP
728 The \fBIere\fR argument to the \fBgsub\fR, \fBmatch\fR, \fBsub\fR functions, and
729 the \fBifs\fR argument to the \fBsplit\fR function (see \fBString Functions\fR)
730 is interpreted as extended regular expressions. These can be either \fBERE\fR
731 tokens or arbitrary expressions, and are interpreted in the same manner as the
732 right-hand side of the \fB~\fR or \fB!~\fR operator.
733 .sp
734 .LP
735 An extended regular expression can be used to separate fields by using the
736 \fB-F\fR \fBIERE\fR option or by assigning a string containing the expression to
737 the built-in variable \fBFS\fR. The default value of the \fBFS\fR variable is a
738 single space character. The following describes \fBFS\fR behavior:
739 .RS +4
740 .TP
741 1.
742 If \fBFS\fR is a single character:
743 .RS +4
744 .TP
745 .ie t \(\bu
746 .el o
747 If \fBFS\fR is the space character, skip leading and trailing blank characters;
748 fields are delimited by sets of one or more blank characters.
749 .RE
750 .RS +4
751 .TP
752 .ie t \(\bu
753 .el o
754 Otherwise, if \fBFS\fR is any other character \fBIc\fR, fields are delimited by
755 each single occurrence of \fBIc\fR.
756 .RE
757 .RE
758 .RS +4
759 .TP
760 2.
761 Otherwise, the string value of \fBFS\fR is considered to be an extended
762 regular expression. Each occurrence of a sequence matching the extended regular
763 expression delimits fields.
764 .RE
765 .sp
766 .LP
767 Except in the \fBgsub\fR, \fBmatch\fR, \fBsplit\fR, and \fBsub\fR built-in
768 functions, regular expression matching is based on input records. That is,
769 record separator characters (the first character of the value of the variable
770 \fBRS\fR, a newline character by default) cannot be embedded in the expression,
771 and no expression matches the record separator character. If the record
772 separator is not a newline character, newline characters embedded in the
773 expression can be matched. In those four built-in functions, regular expression
774 matching are based on text strings. So, any character (including the newline
775 character and the record separator) can be embedded in the pattern and an
776 appropriate pattern matches any character. However, in all \fBnawk\fR regular
777 expression matching, the use of one or more NULL characters in the pattern,
778 input record or text string produces undefined results.
779 .SS "Patterns"
780 .sp

```

```

781 .LP
782 A \fIpattern\fR is any valid \fIexpression,\fR a range specified by two
783 expressions separated by comma, or one of the two special patterns \fBBEGIN\fR
784 or \fBEND\fR.
785 .SS "Special Patterns"
786 .sp
787 .LP
788 The \fBnawk\fR utility recognizes two special patterns, \fBBEGIN\fR and
789 \fBEND\fR. Each \fBBEGIN\fR pattern is matched once and its associated action
790 executed before the first record of input is read (except possibly by use of
791 the \fBgetline\fR function in a prior \fBBEGIN\fR action) and before command
792 line assignment is done. Each \fBEND\fR pattern is matched once and its
793 associated action executed after the last record of input has been read. These
794 two patterns have associated actions.
795 .sp
796 .LP
797 \fBBEGIN\fR and \fBEND\fR do not combine with other patterns. Multiple
798 \fBBEGIN\fR and \fBEND\fR patterns are allowed. The actions associated with the
799 \fBBEGIN\fR patterns are executed in the order specified in the program, as are
800 the \fBEND\fR actions. An \fBEND\fR pattern can precede a \fBBEGIN\fR pattern
801 in a program.
802 .sp
803 .LP
804 If an \fBnawk\fR program consists of only actions with the pattern \fBBEGIN\fR,
805 and the \fBBEGIN\fR action contains no \fBgetline\fR function, \fBnawk\fR exits
806 without reading its input when the last statement in the last \fBBEGIN\fR
807 action is executed. If an \fBnawk\fR program consists of only actions with the
808 pattern \fBEND\fR or only actions with the patterns \fBBEGIN\fR and \fBEND\fR,
809 the input is read before the statements in the \fBEND\fR actions are executed.
810 .SS "Expression Patterns"
811 .sp
812 .LP
813 An expression pattern is evaluated as if it were an expression in a Boolean
814 context. If the result is true, the pattern is considered to match, and the
815 associated action (if any) is executed. If the result is false, the action is
816 not executed.
817 .SS "Pattern Ranges"
818 .sp
819 .LP
820 A pattern range consists of two expressions separated by a comma. In this case,
821 the action is performed for all records between a match of the first expression
822 and the following match of the second expression, inclusive. At this point, the
823 pattern range can be repeated starting at input records subsequent to the end
824 of the matched range.
825 .SS "Actions"
826 .sp
827 .LP
828 An action is a sequence of statements. A statement can be one of the following:
829 .sp
830 .in +2
831 .nf
832 if ( \fIexpression\fR ) \fIstatement\fR [ else \fIstatement\fR ]
833 while ( \fIexpression\fR ) \fIstatement\fR
834 do \fIstatement\fR while ( \fIexpression\fR )
835 for ( \fIexpression\fR ; \fIexpression\fR ; \fIexpression\fR ) \fIstatement\fR
836 for ( \fIivar\fR in \fIarray\fR ) \fIstatement\fR
837 delete \fIarray\fR[\fIsubscript\fR] #delete an array element
838 break
839 continue
840 { [ \fIstatement\fR ] .\|. \. }
841 \fIexpression\fR # commonly variable = expression
842 print [ \fIexpression-list\fR ] [ >\fIexpression\fR ]
843 printf format [ ,\fIexpression-list\fR ] [ >\fIexpression\fR ]
844 next # skip remaining patterns on this input line
845 exit [expr] # skip the rest of the input; exit status is expr
846 return [expr]

```

```

847 .fi
848 .in -2

850 .sp
851 .LP
852 Any single statement can be replaced by a statement list enclosed in braces.
853 The statements are terminated by newline characters or semicolons, and are
854 executed sequentially in the order that they appear.
855 .sp
856 .LP
857 The \fBnext\fR statement causes all further processing of the current input
858 record to be abandoned. The behavior is undefined if a \fBnext\fR statement
859 appears or is invoked in a \fBBEGIN\fR or \fBEND\fR action.
860 .sp
861 .LP
862 The \fBexit\fR statement invokes all \fBEND\fR actions in the order in which
863 they occur in the program source and then terminate the program without reading
864 further input. An \fBexit\fR statement inside an \fBEND\fR action terminates
865 the program without further execution of \fBEND\fR actions. If an expression
866 is specified in an \fBexit\fR statement, its numeric value is the exit status
867 of \fBnawk\fR, unless subsequent errors are encountered or a subsequent
868 \fBexit\fR statement with an expression is executed.
869 .SS "Output Statements"
870 .sp
871 .LP
872 Both \fBprint\fR and \fBprintf\fR statements write to standard output by
873 default. The output is written to the location specified by
874 \fIoutput_redirection\fR if one is supplied, as follows:
875 .sp
876 .in +2
877 .nf
878 \fB>\fR \fIexpression\fR \fB>>\fR \fIexpression\fR \fB|\fR \fIexpression\fR
879 .fi
880 .in -2

882 .sp
883 .LP
884 In all cases, the \fIexpression\fR is evaluated to produce a string that is
885 used as a full pathname to write into (for \fB>\fR or \fB>>\fR) or as a command
886 to be executed (for \fB|\fR). Using the first two forms, if the file of that
887 name is not currently open, it is opened, creating it if necessary and using
888 the first form, truncating the file. The output then is appended to the file.
889 As long as the file remains open, subsequent calls in which \fIexpression\fR
890 evaluates to the same string value simply appends output to the file. The file
891 remains open until the \fBclose\fR function, which is called with an expression
892 that evaluates to the same string value.
893 .sp
894 .LP
895 The third form writes output onto a stream piped to the input of a command. The
896 stream is created if no stream is currently open with the value of
897 \fIexpression\fR as its command name. The stream created is equivalent to one
898 created by a call to the \fBpopen\fR(3C) function with the value of
899 \fIexpression\fR as the \fBcommand\fR argument and a value of \fBw\fR as the
900 \fBmode\fR argument. As long as the stream remains open, subsequent calls in
901 which \fIexpression\fR evaluates to the same string value writes output to the
902 existing stream. The stream remains open until the \fBclose\fR function is
903 called with an expression that evaluates to the same string value. At that
904 time, the stream is closed as if by a call to the \fBpclose\fR function.
905 .sp
906 .LP
907 These output statements take a comma-separated list of \fIexpression\fR \fIis\fR
908 referred in the grammar by the non-terminal symbols \fBexpr_list,\fR
909 \fBprint_expr_list\fR or \fBprint_expr_list_opt.\fR This list is referred to
910 here as the \fIexpression list\fR, and each member is referred to as an
911 \fIexpression argument\fR.
912 .sp

```

913 .LP
 914 The `\fBprint` statement writes the value of each expression argument onto the
 915 indicated output stream separated by the current output field separator (see
 916 variable `\fBOFS` above), and terminated by the output record separator (see
 917 variable `\fBORS` above). All expression arguments is taken as strings, being
 918 converted if necessary; with the exception that the `\fBprintf` format in
 919 `\fBOFMT` is used instead of the value in `\fBCONVFMT`. An empty expression
 920 list stands for the whole input record `\fB(\fR$0\fB)\fR`.
 921 .sp
 922 .LP
 923 The `\fBprintf` statement produces output based on a notation similar to the
 924 File Format Notation used to describe file formats in this document Output is
 925 produced as specified with the first expression argument as the string
 926 `\fBformat` and subsequent expression arguments as the strings `\fBarg1` to
 927 `\fBargn`, inclusive, with the following exceptions:
 928 .RS +4
 929 .TP
 930 1.
 931 The `\fIformat` is an actual character string rather than a graphical
 932 representation. Therefore, it cannot contain empty character positions. The
 933 space character in the `\fIformat` string, in any context other than a
 934 `\fIflag` of a conversion specification, is treated as an ordinary character
 935 that is copied to the output.
 936 .RE
 937 .RS +4
 938 .TP
 939 2.
 940 If the character set contains a Delta character and that character appears
 941 in the `\fIformat` string, it is treated as an ordinary character that is
 942 copied to the output.
 943 .RE
 944 .RS +4
 945 .TP
 946 3.
 947 The `\fIescape` sequences beginning with a backslash character is treated
 948 as sequences of ordinary characters that are copied to the output. Note that
 949 these same sequences is interpreted lexically by `\fBnawk` when they appear in
 950 literal strings, but they is not treated specially by the `\fBprintf`
 951 statement.
 952 .RE
 953 .RS +4
 954 .TP
 955 4.
 956 A `\fIfield width` or `\fIprecision` can be specified as the `\fB*`
 957 character instead of a digit string. In this case the next argument from the
 958 expression list is fetched and its numeric value taken as the field width or
 959 precision.
 960 .RE
 961 .RS +4
 962 .TP
 963 5.
 964 The implementation does not precede or follow output from the `\fBd`
 965 `\fBU` conversion specifications with blank characters not specified by the
 966 `\fIformat` string.
 967 .RE
 968 .RS +4
 969 .TP
 970 6.
 971 The implementation does not precede output from the `\fBo` conversion
 972 specification with leading zeros not specified by the `\fIformat` string.
 973 .RE
 974 .RS +4
 975 .TP
 976 7.
 977 For the `\fBc` conversion specification: if the argument has a numeric
 978 value, the character whose encoding is that value is output. If the value is

979 zero or is not the encoding of any character in the character set, the behavior
 980 is undefined. If the argument does not have a numeric value, the first
 981 character of the string value is output; if the string does not contain any
 982 characters the behavior is undefined.
 983 .RE
 984 .RS +4
 985 .TP
 986 8.
 987 For each conversion specification that consumes an argument, the next
 988 expression argument is evaluated. With the exception of the `\fBc` conversion,
 989 the value is converted to the appropriate type for the conversion
 990 specification.
 991 .RE
 992 .RS +4
 993 .TP
 994 9.
 995 If there are insufficient expression arguments to satisfy all the conversion
 996 specifications in the `\fIformat` string, the behavior is undefined.
 997 .RE
 998 .RS +4
 999 .TP
 1000 10.
 1001 If any character sequence in the `\fIformat` string begins with a %
 1002 character, but does not form a valid conversion specification, the behavior is
 1003 unspecified.
 1004 .RE
 1005 .sp
 1006 .LP
 1007 Both `\fBprint` and `\fBprintf` can output at least `\fB{LINE_MAX}` bytes.
 1008 .SS "Functions"
 1009 .sp
 1010 .LP
 1011 The `\fBnawk` language has a variety of built-in functions: arithmetic,
 1012 string, input/output and general.
 1013 .SS "Arithmetic Functions"
 1014 .sp
 1015 .LP
 1016 The arithmetic functions, except for `\fBint`, are based on the `\fBISO`
 1017 `\fBC` standard. The behavior is undefined in cases where the `\fBISO`
 1018 `\fBC` standard specifies that an error be returned or that the behavior is
 1019 undefined. Although the grammar permits built-in functions to appear with no
 1020 arguments or parentheses, unless the argument or parentheses are indicated as
 1021 optional in the following list (by displaying them within the `\fB[]`
 1022 brackets), such use is undefined.
 1023 .sp
 1024 .ne 2
 1025 .na
 1026 `\fBatan2(\fR\fIy\fR, \fR\fIx\fR)\fR`
 1027 .ad
 1028 .RS 17n
 1029 Return arctangent of `\fR\fIy\fR/\fR\fIx\fR`.
 1030 .RE
 1031 .sp
 1032 .sp
 1033 .ne 2
 1034 .na
 1035 `\fBacos(\fR\fIx\fR)\fR`
 1036 .ad
 1037 .RS 17n
 1038 Return cosine of `\fR\fIx\fR` where `\fR\fIx\fR` is in radians.
 1039 .RE
 1040 .sp
 1041 .sp
 1042 .ne 2
 1043 .na
 1044 `\fBasin(\fR\fIx\fR)\fR`

```

1045 .ad
1046 .RS 17n
1047 Return sine of \fIx,\fR where \fIx\fR is in radians.
1048 .RE

1050 .sp
1051 .ne 2
1052 .na
1053 \fB\fBexp\fR(\fIx\fR)\fR
1054 .ad
1055 .RS 17n
1056 Return the exponential function of \fIx\fR.
1057 .RE

1059 .sp
1060 .ne 2
1061 .na
1062 \fB\fBlog\fR(\fIx\fR)\fR
1063 .ad
1064 .RS 17n
1065 Return the natural logarithm of \fIx\fR.
1066 .RE

1068 .sp
1069 .ne 2
1070 .na
1071 \fB\fBsqrt\fR(\fIx\fR)\fR
1072 .ad
1073 .RS 17n
1074 Return the square root of \fIx\fR.
1075 .RE

1077 .sp
1078 .ne 2
1079 .na
1080 \fB\fBint\fR(\fIx\fR)\fR
1081 .ad
1082 .RS 17n
1083 Truncate its argument to an integer. It is truncated toward 0 when \fIx\fR > 0.
1084 .RE

1086 .sp
1087 .ne 2
1088 .na
1089 \fB\fBrand()\fR\fR
1090 .ad
1091 .RS 17n
1092 Return a random number \fIn\fR, such that 0 \(<= \fIn\fR < 1.
1093 .RE

1095 .sp
1096 .ne 2
1097 .na
1098 \fB\fBsrnd\fR([\fBexpr\fR])\fR
1099 .ad
1100 .RS 17n
1101 Set the seed value for \fBrand\fR to \fIexpr\fR or use the time of day if
1102 \fIexpr\fR is omitted. The previous seed value is returned.
1103 .RE

1105 .SS "String Functions"
1106 .sp
1107 .LP
1108 The string functions in the following list shall be supported. Although the
1109 grammar permits built-in functions to appear with no arguments or parentheses,
1110 unless the argument or parentheses are indicated as optional in the following

```

```

1111 list (by displaying them within the \fB[ ]\fR brackets), such use is undefined.
1112 .sp
1113 .ne 2
1114 .na
1115 \fB\fBgsub\fR(\fIere\fR,\fIrepl\fR[,|\fIin\fR])\fR
1116 .ad
1117 .sp .6
1118 .RS 4n
1119 Behave like \fBsub\fR (see below), except that it replaces all occurrences of
1120 the regular expression (like the \fBed\fR utility global substitute) in
1121 \fB$0\fR or in the \fIin\fR argument, when specified.
1122 .RE

1124 .sp
1125 .ne 2
1126 .na
1127 \fB\fBindex\fR(\fIs\fR,\fIt\fR)\fR
1128 .ad
1129 .sp .6
1130 .RS 4n
1131 Return the position, in characters, numbering from 1, in string \fIs\fR where
1132 string \fIt\fR first occurs, or zero if it does not occur at all.
1133 .RE

1135 .sp
1136 .ne 2
1137 .na
1138 \fB\fBlength\fR([\fIs\fR])\fR
1139 .ad
1140 .sp .6
1141 .RS 4n
1142 Return the length, in characters, of its argument taken as a string, or of the
1143 whole record, \fB$0\fR, if there is no argument.
1144 .RE

1146 .sp
1147 .ne 2
1148 .na
1149 \fB\fBmatch\fR(\fIs\fR,\fIere\fR)\fR
1150 .ad
1151 .sp .6
1152 .RS 4n
1153 Return the position, in characters, numbering from 1, in string \fIs\fR where
1154 the extended regular expression \fIere\fR occurs, or zero if it does not occur
1155 at all. \fBRSSTART\fR is set to the starting position (which is the same as the
1156 returned value), zero if no match is found; \fBRLLENGTH\fR is set to the length
1157 of the matched string, \(\mil if no match is found.
1158 .RE

1160 .sp
1161 .ne 2
1162 .na
1163 \fB\fBsplit\fR(\fIs\fR,\fIa\fR[,|\fIifs\fR])\fR
1164 .ad
1165 .sp .6
1166 .RS 4n
1167 Split the string \fIs\fR into array elements \fIa\fR[1], \fIa\fR[2],
1168 \fB&...\fR \fIa\fR[\fIn\fR], and return \fIn\fR. The separation is done with
1169 the extended regular expression \fIifs\fR or with the field separator \fBFS\fR
1170 if \fIifs\fR is not given. Each array element has a string value when created.
1171 If the string assigned to any array element, with any occurrence of the
1172 decimal-point character from the current locale changed to a period character,
1173 would be considered a \fInumeric string\fR; the array element also has the
1174 numeric value of the \fInumeric string\fR. The effect of a null string as the
1175 value of \fIifs\fR is unspecified.
1176 .RE

```

```

1178 .sp
1179 .ne 2
1180 .na
1181 \fB\fBsprintf\fR(\fBfmt\fR,\fIexpr\fR,\fIexpr\fR,\fB&...\fR)\fR
1182 .ad
1183 .sp .6
1184 .RS 4n
1185 Format the expressions according to the \fBprintf\fR format given by \fIfmt\fR
1186 and return the resulting string.
1187 .RE

1189 .sp
1190 .ne 2
1191 .na
1192 \fB\fBsub\fR(\fIere\fR,\fIrepl\fR[,|\fIin\fR])\fR
1193 .ad
1194 .sp .6
1195 .RS 4n
1196 Substitute the string \fIrepl\fR in place of the first instance of the extended
1197 regular expression \fBERE\fR in string in and return the number of
1198 substitutions. An ampersand ( \fB&\fR ) appearing in the string \fIrepl\fR is
1199 replaced by the string from in that matches the regular expression. An
1200 ampersand preceded with a backslash ( \fB\e\fR ) is interpreted as the literal
1201 ampersand character. An occurrence of two consecutive backslashes is
1202 interpreted as just a single literal backslash character. Any other occurrence
1203 of a backslash (for example, preceding any other character) is treated as a
1204 literal backslash character. If \fIrepl\fR is a string literal, the handling of
1205 the ampersand character occurs after any lexical processing, including any
1206 lexical backslash escape sequence processing. If \fBin\fR is specified and it
1207 is not an \fB\value\fR the behavior is undefined. If in is omitted, \fBnawk\fR
1208 uses the current record (\fB$0\fR) in its place.
1209 .RE

1211 .sp
1212 .ne 2
1213 .na
1214 \fB\fBsubstr\fR(\fIis\fR,\fIm\fR[,|\fIn\fR])\fR
1215 .ad
1216 .sp .6
1217 .RS 4n
1218 Return the at most \fIn\fR-character substring of \fIis\fR that begins at
1219 position \fIm,\fR numbering from 1. If \fIn\fR is missing, the length of the
1220 substring is limited by the length of the string \fIis\fR.
1221 .RE

1223 .sp
1224 .ne 2
1225 .na
1226 \fB\fBtolower\fR(\fIis\fR)\fR
1227 .ad
1228 .sp .6
1229 .RS 4n
1230 Return a string based on the string \fIis\fR. Each character in \fIis\fR that is
1231 an upper-case letter specified to have a \fBtolower\fR mapping by the
1232 \fBLC_CTYPE\fR category of the current locale is replaced in the returned
1233 string by the lower-case letter specified by the mapping. Other characters in
1234 \fIis\fR are unchanged in the returned string.
1235 .RE

1237 .sp
1238 .ne 2
1239 .na
1240 \fB\fBtoupper\fR(\fIis\fR)\fR
1241 .ad
1242 .sp .6

```

```

1243 .RS 4n
1244 Return a string based on the string \fIis\fR. Each character in \fIis\fR that is
1245 a lower-case letter specified to have a \fBtoupper\fR mapping by the
1246 \fBLC_CTYPE\fR category of the current locale is replaced in the returned
1247 string by the upper-case letter specified by the mapping. Other characters in
1248 \fIis\fR are unchanged in the returned string.
1249 .RE

1251 .sp
1252 .LP
1253 All of the preceding functions that take \fIERE\fR as a parameter expect a
1254 pattern or a string valued expression that is a regular expression as defined
1255 below.
1256 .SS "Input/Output and General Functions"
1257 .sp
1258 .LP
1259 The input/output and general functions are:
1260 .sp
1261 .ne 2
1262 .na
1263 \fB\fBclose(\fR\fIexpression\fR)\fR
1264 .ad
1265 .RS 27n
1266 Close the file or pipe opened by a \fBprint\fR or \fBprintf\fR statement or a
1267 call to \fBgetline\fR with the same string-valued \fIexpression\fR. If the
1268 close was successful, the function returns \fB0\fR; otherwise, it returns
1269 non-zero.
1270 .RE

1272 .sp
1273 .ne 2
1274 .na
1275 \fB\fB\fIexpression\fR|\fBgetline\fR[\fIivar\fR]\fR
1276 .ad
1277 .RS 27n
1278 Read a record of input from a stream piped from the output of a command. The
1279 stream is created if no stream is currently open with the value of
1280 \fIexpression\fR as its command name. The stream created is equivalent to one
1281 created by a call to the \fBpopen\fR function with the value of
1282 \fIexpression\fR as the \fIcommand\fR argument and a value of \fB\r\fR as the
1283 \fImode\fR argument. As long as the stream remains open, subsequent calls in
1284 which \fIexpression\fR evaluates to the same string value reads subsequent
1285 records from the file. The stream remains open until the \fBclose\fR function
1286 is called with an expression that evaluates to the same string value. At that
1287 time, the stream is closed as if by a call to the \fBpclose\fR function. If
1288 \fIivar\fR is missing, \fB$0\fR and \fBNF\fR is set. Otherwise, \fIivar\fR is
1289 set.
1290 .sp
1291 The \fBgetline\fR operator can form ambiguous constructs when there are
1292 operators that are not in parentheses (including concatenate) to the left of
1293 the \fB|\fR (to the beginning of the expression containing \fBgetline\fR). In
1294 the context of the \fB$\fR operator, \fB|\fR behaves as if it had a lower
1295 precedence than \fB$\fR. The result of evaluating other operators is
1296 unspecified, and all such uses of portable applications must be put in
1297 parentheses properly.
1298 .RE

1300 .sp
1301 .ne 2
1302 .na
1303 \fB\fB\fIexpression\fR|\fR
1304 .ad
1305 .RS 30n
1306 Set \fB$0\fR to the next input record from the current input file. This form of
1307 \fBgetline\fR sets the \fBNF\fR, \fBNR\fR, and \fBFNR\fR variables.
1308 .RE

```

1310 .sp
 1311 .ne 2
 1312 .na
 1313 \fB\fBgetline\fR \fIivar\fR\fR
 1314 .ad
 1315 .RS 30n
 1316 Set variable \fIivar\fR to the next input record from the current input file.
 1317 This form of \fBgetline\fR sets the \fBFRNR\fR and \fBFRNR\fR variables.
 1318 .RE

1320 .sp
 1321 .ne 2
 1322 .na
 1323 \fB\fBgetline\fR [\fIivar\fR] \fB<\fR \fIexpression\fR\fR
 1324 .ad
 1325 .RS 30n
 1326 Read the next record of input from a named file. The \fIexpression\fR is
 1327 evaluated to produce a string that is used as a full pathname. If the file of
 1328 that name is not currently open, it is opened. As long as the stream remains
 1329 open, subsequent calls in which \fIexpression\fR evaluates to the same string
 1330 value reads subsequent records from the file. The file remains open until the
 1331 \fBfclose\fR function is called with an expression that evaluates to the same
 1332 string value. If \fIivar\fR is missing, \fB\$0\fR and \fBFRNR\fR is set. Otherwise,
 1333 \fIivar\fR is set.
 1334 .sp
 1335 The \fBgetline\fR operator can form ambiguous constructs when there are binary
 1336 operators that are not in parentheses (including concatenate) to the right of
 1337 the \fB<\fR (up to the end of the expression containing the \fBgetline\fR). The
 1338 result of evaluating such a construct is unspecified, and all such uses of
 1339 portable applications must be put in parentheses properly.
 1340 .RE

1342 .sp
 1343 .ne 2
 1344 .na
 1345 \fB\fBsystem\fR(\fIexpression\fR)\fR
 1346 .ad
 1347 .RS 30n
 1348 Execute the command given by \fIexpression\fR in a manner equivalent to the
 1349 \fBsystem\fR(3C) function and return the exit status of the command.
 1350 .RE

1352 .sp
 1353 .LP
 1354 All forms of \fBgetline\fR return \fB1\fR for successful input, \fB0\fR for end
 1355 of file, and \fB(mil)\fR for an error.
 1356 .sp
 1357 .LP
 1358 Where strings are used as the name of a file or pipeline, the strings must be
 1359 textually identical. The terminology "same string value" implies that
 1360 "equivalent strings", even those that differ only by space characters,
 1361 represent different files.
 1362 .SS "User-defined Functions"
 1363 .sp
 1364 .LP
 1365 The \fBnawk\fR language also provides user-defined functions. Such functions
 1366 can be defined as:
 1367 .sp
 1368 .in +2
 1369 .nf
 1370 \fBfunction\fR \fIiname\fR(\fIargs\fR, \fI.\fR, \fI.\fR, \fI.\fR) { \fIstatements\fR }
 1371 .fi
 1372 .in -2

1374 .sp

1375 .LP
 1376 A function can be referred to anywhere in an \fBnawk\fR program; in particular,
 1377 its use can precede its definition. The scope of a function is global.
 1378 .sp
 1379 .LP
 1380 Function arguments can be either scalars or arrays; the behavior is undefined
 1381 if an array name is passed as an argument that the function uses as a scalar,
 1382 or if a scalar expression is passed as an argument that the function uses as an
 1383 array. Function arguments are passed by value if scalar and by reference if
 1384 array name. Argument names are local to the function; all other variable names
 1385 are global. The same name is not used as both an argument name and as the name
 1386 of a function or a special \fBnawk\fR variable. The same name must not be used
 1387 both as a variable name with global scope and as the name of a function. The
 1388 same name must not be used within the same scope both as a scalar variable and
 1389 as an array.
 1390 .sp
 1391 .LP
 1392 The number of parameters in the function definition need not match the number
 1393 of parameters in the function call. Excess formal parameters can be used as
 1394 local variables. If fewer arguments are supplied in a function call than are in
 1395 the function definition, the extra parameters that are used in the function
 1396 body as scalars are initialized with a string value of the null string and a
 1397 numeric value of zero, and the extra parameters that are used in the function
 1398 body as arrays are initialized as empty arrays. If more arguments are supplied
 1399 in a function call than are in the function definition, the behavior is
 1400 undefined.
 1401 .sp
 1402 .LP
 1403 When invoking a function, no white space can be placed between the function
 1404 name and the opening parenthesis. Function calls can be nested and recursive
 1405 calls can be made upon functions. Upon return from any nested or recursive
 1406 function call, the values of all of the calling function's parameters are
 1407 unchanged, except for array parameters passed by reference. The \fBreturn\fR
 1408 statement can be used to return a value. If a \fBreturn\fR statement appears
 1409 outside of a function definition, the behavior is undefined.
 1410 .sp
 1411 .LP
 1412 In the function definition, newline characters are optional before the opening
 1413 brace and after the closing brace. Function definitions can appear anywhere in
 1414 the program where a \fBpattern-action\fR pair is allowed.
 1415 .SH USAGE
 1416 .sp
 1417 .LP
 1418 The \fBindex\fR, \fBlength\fR, \fBmatch\fR, and \fBsubstr\fR functions should
 1419 not be confused with similar functions in the \fBISO C\fR standard; the
 1420 \fBnawk\fR versions deal with characters, while the \fBISO C\fR standard deals
 1421 with bytes.
 1422 .sp
 1423 .LP
 1424 Because the concatenation operation is represented by adjacent expressions
 1425 rather than an explicit operator, it is often necessary to use parentheses to
 1426 enforce the proper evaluation precedence.
 1427 .sp
 1428 .LP
 1429 See \fBlargefile\fR(5) for the description of the behavior of \fBnawk\fR when
 1430 encountering files greater than or equal to 2 Gbyte (2³¹ bytes).
 1431 .SH EXAMPLES
 1432 .sp
 1433 .LP
 1434 The \fBnawk\fR program specified in the command line is most easily specified
 1435 within single-quotes (for example, \fB&'\fR\fIprogram\fR\fB&'\fR) for
 1436 applications using \fBsh\fR, because \fBnawk\fR programs commonly contain
 1437 characters that are special to the shell, including double-quotes. In the cases
 1438 where a \fBnawk\fR program contains single-quote characters, it is usually
 1439 easiest to specify most of the program as strings within single-quotes
 1440 concatenated by the shell with quoted single-quote characters. For example:

```

1441 .sp
1442 .in +2
1443 .nf
1444 nawk '/'\e'/' { print "quote:", $0 }'
1445 .fi
1446 .in -2

1448 .sp
1449 .LP
1450 prints all lines from the standard input containing a single-quote character,
1451 prefixed with \fBquote:\fR.
1452 .sp
1453 .LP
1454 The following are examples of simple \fBnawk\fR programs:
1455 .LP
1456 \fBExample 1 \fRWrite to the standard output all input lines for which field 3
1457 is greater than 5:
1458 .sp
1459 .in +2
1460 .nf
1461 \fB$3 > 5\fR
1462 .fi
1463 .in -2
1464 .sp

1466 .LP
1467 \fBExample 2 \fRWrite every tenth line:
1468 .sp
1469 .in +2
1470 .nf
1471 \fB(NR % 10) == 0\fR
1472 .fi
1473 .in -2
1474 .sp

1476 .LP
1477 \fBExample 3 \fRWrite any line with a substring matching the regular
1478 expression:
1479 .sp
1480 .in +2
1481 .nf
1482 \fB/(G|D)(2[0-9][[:alpha:]]*)\fR
1483 .fi
1484 .in -2
1485 .sp

1487 .LP
1488 \fBExample 4 \fRPrint any line with a substring containing a G or D, followed
1489 by a sequence of digits and characters:
1490 .sp
1491 .LP
1492 This example uses character classes \fBdigit\fR and \fBalpha\fR to match
1493 language-independent digit and alphabetic characters, respectively.

1495 .sp
1496 .in +2
1497 .nf
1498 \fB/(G|D)([[:digit:]][[:alpha:]]*)\fR
1499 .fi
1500 .in -2
1501 .sp

1503 .LP
1504 \fBExample 5 \fRWrite any line in which the second field matches the regular
1505 expression and the fourth field does not:
1506 .sp

```

```

1507 .in +2
1508 .nf
1509 \fB$2 ~ /xyz/ && $4 !~ /xyz/\fR
1510 .fi
1511 .in -2
1512 .sp

1514 .LP
1515 \fBExample 6 \fRWrite any line in which the second field contains a backslash:
1516 .sp
1517 .in +2
1518 .nf
1519 \fB$2 ~ /\e/e/\fR
1520 .fi
1521 .in -2
1522 .sp

1524 .LP
1525 \fBExample 7 \fRWrite any line in which the second field contains a backslash
1526 (alternate method):
1527 .sp
1528 .LP
1529 Notice that backslash escapes are interpreted twice, once in lexical processing
1530 of the string and once in processing the regular expression.

1532 .sp
1533 .in +2
1534 .nf
1535 \fB$2 ~ "\e\e\e\e"\fR
1536 .fi
1537 .in -2
1538 .sp

1540 .LP
1541 \fBExample 8 \fRWrite the second to the last and the last field in each line,
1542 separating the fields by a colon:
1543 .sp
1544 .in +2
1545 .nf
1546 \fB{OFS=":";print $(NF-1), $NF}\fR
1547 .fi
1548 .in -2
1549 .sp

1551 .LP
1552 \fBExample 9 \fRWrite the line number and number of fields in each line:
1553 .sp
1554 .LP
1555 The three strings representing the line number, the colon and the number of
1556 fields are concatenated and that string is written to standard output.

1558 .sp
1559 .in +2
1560 .nf
1561 \fB{print NR ":" NF}\fR
1562 .fi
1563 .in -2
1564 .sp

1566 .LP
1567 \fBExample 10 \fRWrite lines longer than 72 characters:
1568 .sp
1569 .in +2
1570 .nf
1571 \fB[length($0) > 72]\fR
1572 .fi

```

```

1573 .in -2
1574 .sp

1576 .LP
1577 \fBExample 11 \fRWrite first two fields in opposite order separated by the OFS:
1578 .sp
1579 .in +2
1580 .nf
1581 \fB{ print $2, $1 }\fR
1582 .fi
1583 .in -2
1584 .sp

1586 .LP
1587 \fBExample 12 \fRSame, with input fields separated by comma or space and tab
1588 characters, or both:
1589 .sp
1590 .in +2
1591 .nf
1592 \fBBEGIN { FS = ",[\et]*|[\et]+" }
1593     { print $2, $1 }\fR
1594 .fi
1595 .in -2
1596 .sp

1598 .LP
1599 \fBExample 13 \fRAdd up first column, print sum and average:
1600 .sp
1601 .in +2
1602 .nf
1603 \fB{s += $1 }
1604 END {print "sum is ", s, " average is", s/NR}\fR
1605 .fi
1606 .in -2
1607 .sp

1609 .LP
1610 \fBExample 14 \fRWrite fields in reverse order, one per line (many lines out
1611 for each line in):
1612 .sp
1613 .in +2
1614 .nf
1615 \fB{ for (i = NF; i > 0; --i) print $i }\fR
1616 .fi
1617 .in -2
1618 .sp

1620 .LP
1621 \fBExample 15 \fRWrite all lines between occurrences of the strings "start" and
1622 "stop":
1623 .sp
1624 .in +2
1625 .nf
1626 \fB/start/, /stop/>\fR
1627 .fi
1628 .in -2
1629 .sp

1631 .LP
1632 \fBExample 16 \fRWrite all lines whose first field is different from the
1633 previous one:
1634 .sp
1635 .in +2
1636 .nf
1637 \fB$1 != prev { print; prev = $1 }\fR
1638 .fi

```

```

1639 .in -2
1640 .sp

1642 .LP
1643 \fBExample 17 \fRSimulate the echo command:
1644 .sp
1645 .in +2
1646 .nf
1647 \fBBEGIN {
1648     for (i = 1; i < ARGV; ++i)
1649         printf "%s%s", ARGV[i], i==ARGV-1?"\n":""
1650     }\fR
1651 .fi
1652 .in -2
1653 .sp

1655 .LP
1656 \fBExample 18 \fRWrite the path prefixes contained in the PATH environment
1657 variable, one per line:
1658 .sp
1659 .in +2
1660 .nf
1661 \fBBEGIN {
1662     n = split (ENVIRON["PATH"], path, ":")
1663     for (i = 1; i <= n; ++i)
1664         print path[i]
1665     }\fR
1666 .fi
1667 .in -2
1668 .sp

1670 .LP
1671 \fBExample 19 \fRPrint the file "input", filling in page numbers starting at 5:
1672 .sp
1673 .LP
1674 If there is a file named \fBinput\fR containing page headers of the form

1676 .sp
1677 .in +2
1678 .nf
1679 Page#
1680 .fi
1681 .in -2

1683 .sp
1684 .LP
1685 and a file named \fBprogram\fR that contains

1687 .sp
1688 .in +2
1689 .nf
1690 /Page/{ $2 = n++; }
1691 { print }
1692 .fi
1693 .in -2

1695 .sp
1696 .LP
1697 then the command line

1699 .sp
1700 .in +2
1701 .nf
1702 \fBnawk -f program n=5 input\fR
1703 .fi
1704 .in -2

```



```

1705 .sp
1707 .sp
1708 .LP
1709 prints the file \fBinput\fR, filling in page numbers starting at 5.

1711 .SH ENVIRONMENT VARIABLES
1712 .sp
1713 .LP
1714 See \fBenvron\fR(5) for descriptions of the following environment variables
1715 that affect execution: \fBLC_COLLATE\fR, \fBLC_CTYPE\fR, \fBLC_MESSAGES\fR, and
1716 \fBNLSPATH\fR.
1717 .sp
1718 .ne 2
1719 .na
1720 \fBLC_NUMERIC\fR
1721 .ad
1722 .RS 14n
1723 Determine the radix character used when interpreting numeric input, performing
1724 conversions between numeric and string values and formatting numeric output.
1725 Regardless of locale, the period character (the decimal-point character of the
1726 POSIX locale) is the decimal-point character recognized in processing \fBawk\fR
1727 programs (including assignments in command-line arguments).
1728 .RE

1730 .SH EXIT STATUS
1731 .sp
1732 .LP
1733 The following exit values are returned:
1734 .sp
1735 .ne 2
1736 .na
1737 \fB0\fR
1738 .ad
1739 .RS 6n
1740 All input files were processed successfully.
1741 .RE

1743 .sp
1744 .ne 2
1745 .na
1746 \fB>0\fR
1747 .ad
1748 .RS 6n
1749 An error occurred.
1750 .RE

1752 .sp
1753 .LP
1754 The exit status can be altered within the program by using an \fBexit\fR
1755 expression.

1757 .SH SEE ALSO
1758 .sp
1759 .LP
1760 \fBawk\fR(1), \fBed\fR(1), \fBegrep\fR(1), \fBgrep\fR(1), \fBlex\fR(1),
1761 \fBsed\fR(1), \fBpopen\fR(3C), \fBprintf\fR(3C), \fBsystem\fR(3C),
1762 \fBattributes\fR(5), \fBenvron\fR(5), \fBlargefile\fR(5), \fBregex\fR(5),
1763 \fBxpg4\fR(5)
1764 .sp
1765 .LP
1766 Aho, A. V., B. W. Kernighan, and P. J. Weinberger, \fIThe AWK Programming
1767 Language\fR, Addison-Wesley, 1988.
1768 .SH DIAGNOSTICS
1769 .sp
1770 .LP

```

```

1771 If any \fIfile\fR operand is specified and the named file cannot be accessed,
1772 \fBnawk\fR writes a diagnostic message to standard error and terminate without
1773 any further action.
1774 .sp
1775 .LP
1776 If the program specified by either the \fIprogram\fR operand or a
1777 \fIprogfile\fR operand is not a valid \fBnawk\fR program (as specified in
1778 \fBEXTENDED DESCRIPTION\fR), the behavior is undefined.
1779 .SH NOTES
1780 .sp
1781 .LP
1782 Input white space is not preserved on output if fields are involved.
1783 .sp
1784 .LP
1785 There are no explicit conversions between numbers and strings. To force an
1786 expression to be treated as a number add 0 to it; to force it to be treated as
1787 a string concatenate the null string (\fB"\fR) to it.

```

10149 Sun May 26 18:23:36 2013

new/usr/src/man/man1m/cpustat.1m

XXX Remove awk(1)

```

1  \' te
2  .\" Copyright (c) 2005, Sun Microsystems, Inc.
3  .TH CPUSTAT 1M "Jun 16, 2009"
4  .SH NAME
5  cpustat \- monitor system behavior using CPU performance counters
6  .SH SYNOPSIS
7  .LP
8  .nf
9  \fBcpustat\fR \fB-c\fR \fIeventspec\fR [\fB-c\fR \fIeventspec\fR]... [\fB-p\fR \
10 [\fB-sntD\fR] \fIinterval\fR [\fIcount\fR]]
11 .fi

13 .LP
14 .nf
15 \fBcpustat\fR \fB-h\fR
16 .fi

18 .SH DESCRIPTION
19 .sp
20 .LP
21 The \fBcpustat\fR utility allows \fBCPU\fR performance counters to be used to
22 monitor the overall behavior of the \fBCPU\fRs in the system.
23 .sp
24 .LP
25 If \fIinterval\fR is specified, \fBcpustat\fR samples activity every
26 \fIinterval\fR seconds, repeating forever. If a \fIcount\fR is specified, the
27 statistics are repeated \fIcount\fR times. If neither are specified, an
28 interval of five seconds is used, and there is no limit to the number of
29 samples that are taken.
30 .SH OPTIONS
31 .sp
32 .LP
33 The following options are supported:
34 .sp
35 .ne 2
36 .na
37 \fB-c\fR \fIeventspec\fR
38 .ad
39 .sp .6
40 .RS 4n
41 Specifies a set of events for the \fBCPU\fR performance counters to monitor.
42 The syntax of these event specifications is:
43 .sp
44 .in +2
45 .nf
46 [picn=]\fIeventn\fR[,attr[\fIn\fR][=\fIval\fR]][, [picn=]\fIeventn\fR
47 [,attr[n][=\fIval\fR]],...,]
48 .fi
49 .in -2
50 .sp

52 You can use the \fB-h\fR option to obtain a list of available events and
53 attributes. This causes generation of the usage message. You can omit an
54 explicit counter assignment, in which case \fBcpustat\fR attempts to choose a
55 capable counter automatically.
56 .sp
57 Attribute values can be expressed in hexadecimal, octal, or decimal notation,
58 in a format suitable for \fBstrtoll\fR(3C). An attribute present in the event
59 specification without an explicit value receives a default value of \fB1\fR. An
60 attribute without a corresponding counter number is applied to all counters in
61 the specification.

```

```

62 .sp
63 The semantics of these event specifications can be determined by reading the
64 \fBCPU\fR manufacturer's documentation for the events.
65 .sp
66 Multiple \fB-c\fR options can be specified, in which case the command cycles
67 between the different event settings on each sample.
68 .RE

70 .sp
71 .ne 2
72 .na
73 \fB-D\fR
74 .ad
75 .sp .6
76 .RS 4n
77 Enables debug mode.
78 .RE

80 .sp
81 .ne 2
82 .na
83 \fB-h\fR
84 .ad
85 .sp .6
86 .RS 4n
87 Prints an extensive help message on how to use the utility and how to program
88 the processor-dependent counters.
89 .RE

91 .sp
92 .ne 2
93 .na
94 \fB-n\fR
95 .ad
96 .sp .6
97 .RS 4n
98 Omits all header output (useful if \fBcpustat\fR is the beginning of a
99 pipeline).
100 .RE

102 .sp
103 .ne 2
104 .na
105 \fB-p\fR \fIperiod\fR
106 .ad
107 .sp .6
108 .RS 4n
109 Causes \fBcpustat\fR to cycle through the list of \fIeventspec\fRs every
110 \fIperiod\fR seconds. The tool sleeps after each cycle until \fIperiod\fR
111 seconds have elapsed since the first \fIeventspec\fR was measured.
112 .sp
113 When this option is present, the optional \fIcount\fR parameter specifies the
114 number of total cycles to make (instead of the number of total samples to
115 take). If \fIperiod\fR is less than the number of \fIeventspec\fRs times
116 \fIinterval\fR, the tool acts as if period is \fB0\fR.
117 .RE

119 .sp
120 .ne 2
121 .na
122 \fB-s\fR
123 .ad
124 .sp .6
125 .RS 4n
126 Creates an idle soaker thread to spin while system-only \fIeventspec\fRs are
127 bound. One idle soaker thread is bound to each CPU in the current processor

```

```

128 set. System-only \fIeventspec\fRs contain both the \fBnouser\fR and the
129 \fBsys\fR tokens and measure events that occur while the CPU is operating in
130 privileged mode. This option prevents the kernel's idle loop from running and
131 triggering system-mode events.
132 .RE

134 .sp
135 .ne 2
136 .na
137 \fB\fB-T\fR \fBu\fR | \fBd\fR\fR
138 .ad
139 .sp .6
140 .RS 4n
141 Display a time stamp.
142 .sp
143 Specify \fBu\fR for a printed representation of the internal representation of
144 time. See \fBtime\fR(2). Specify \fBd\fR for standard date format. See
145 \fBdate\fR(1).
146 .RE

148 .sp
149 .ne 2
150 .na
151 \fB\fB-t\fR\fR
152 .ad
153 .sp .6
154 .RS 4n
155 Prints an additional column of processor cycle counts, if available on the
156 current architecture.
157 .RE

159 .SH USAGE
160 .sp
161 .LP
162 A closely related utility, \fBcputrack\fR(1), can be used to monitor the
163 behavior of individual applications with little or no interference from other
164 activities on the system.
165 .sp
166 .LP
167 The \fBcpustat\fR utility must be run by the super-user, as there is an
168 intrinsic conflict between the use of the \fBCPU\fR performance counters
169 system-wide by \fBcpustat\fR and the use of the \fBCPU\fR performance counters
170 to monitor an individual process (for example, by \fBcputrack\fR.)
171 .sp
172 .LP
173 Once any instance of this utility has started, no further per-process or
174 per-\fBLWP\fR use of the counters is allowed until the last instance of the
175 utility terminates.
176 .sp
177 .LP
178 The times printed by the command correspond to the wallclock time when the
179 hardware counters were actually sampled, instead of when the program told the
180 kernel to sample them. The time is derived from the same timebase as
181 \fBgethrtime\fR(3C).
182 .sp
183 .LP
184 The processor cycle counts enabled by the \fB-t\fR option always apply to both
185 user and system modes, regardless of the settings applied to the performance
186 counter registers.
187 .sp
188 .LP
189 On some hardware platforms running in system mode using the "sys" token, the
190 counters are implemented using 32-bit registers. While the kernel attempts to
191 catch all overflows to synthesize 64-bit counters, because of hardware
192 implementation restrictions, overflows can be lost unless the sampling interval
193 is kept short enough. The events most prone to wrap are those that count

```

```

194 processor clock cycles. If such an event is of interest, sampling should occur
195 frequently so that less than 4 billion clock cycles can occur between samples.
196 .sp
197 .LP
198 The output of cpustat is designed to be readily parseable by \fBawk\fR(1) and
199 The output of cpustat is designed to be readily parseable by \fBnawk\fR(1) and
200 \fBperl\fR(1), thereby allowing performance tools to be composed by embedding
201 \fBcpustat\fR in scripts. Alternatively, tools can be constructed directly
202 using the same \fBAPI\fRs that \fBcpustat\fR is built upon using the facilities
203 of \fBlibcpc\fR(3LIB). See \fBcpc\fR(3CPC).
204 .sp
205 .LP
206 The \fBcpustat\fR utility only monitors the \fBCPU\fRs that are accessible to
207 it in the current processor set. Thus, several instances of the utility can be
208 running on the \fBCPU\fRs in different processor sets. See \fBprrset\fR(1M) for
209 more information about processor sets.
210 .sp
211 .LP
212 Because \fBcpustat\fR uses \fBLWP\fRs bound to \fBCPU\fRs, the utility might
213 have to be terminated before the configuration of the relevant processor can be
214 changed.
215 .SH EXAMPLES
216 .SS "SPARC"
217 .LP
218 \fBExample 1 \fRMeasuring External Cache References and Misses
219 .sp
220 The following example measures misses and references in the external cache.
221 These occur while the processor is operating in user mode on an UltraSPARC
222 machine.

224 .sp
225 .in +2
226 .nf
227 example% cpustat -c EC_ref,EC_misses 1 3

229     time cpu event      pic0      pic1
230     1.008  0 tick        69284     1647
231     1.008  1 tick        43284     1175
232     2.008  0 tick       179576     1834
233     2.008  1 tick       202022    12046
234     3.008  0 tick        93262      384
235     3.008  1 tick        63649     1118
236     3.008  2 total       651077    18204
237 .fi
238 .in -2
239 .sp

241 .SS "x86"
242 .LP
243 \fBExample 2 \fRMeasuring Branch Prediction Success on Pentium 4
244 .sp
245 .LP
246 The following example measures branch mispredictions and total branch
247 instructions in user and system mode on a Pentium 4 machine.

249 .sp
250 .in +2
251 .nf
252 example% cpustat -c \e
253     pic12=branch_retired,emask12=0x4,pic14=branch_retired,\e
254     emask14=0xf,sys 1 3

256     time cpu event      pic12     pic14
257     1.010  1 tick         458       684
258     1.010  0 tick         305       511

```

```

259 2.010 0 tick 181 269
260 2.010 1 tick 469 684
261 3.010 0 tick 182 269
262 3.010 1 tick 468 684
263 3.010 2 total 2063 3101
264 .fi
265 .in -2
266 .sp

268 .LP
269 \fBExample 3 \fRCounting Memory Accesses on Opteron
270 .sp
271 .LP
272 The following example determines the number of memory accesses made through
273 each memory controller on an Opteron, broken down by internal memory latency:

275 .sp
276 .in +2
277 .nf
278 cpustat -c \e
279 pic0=NB_mem_ctrlr_page_access,umask0=0x01, \e
280 pic1=NB_mem_ctrlr_page_access,umask1=0x02, \e
281 pic2=NB_mem_ctrlr_page_access,umask2=0x04,sys \e
282 1

284 time cpu event pic0 pic1 pic2
285 1.003 0 tick 41976 53519 7720
286 1.003 1 tick 5589 19402 731
287 2.003 1 tick 6011 17005 658
288 2.003 0 tick 43944 45473 7338
289 3.003 1 tick 7105 20177 762
290 3.003 0 tick 47045 48025 7119
291 4.003 0 tick 43224 46296 6694
292 4.003 1 tick 5366 19114 652
293 .fi
294 .in -2
295 .sp

297 .SH WARNINGS
298 .sp
299 .LP
300 By running the \fBcpustat\fR command, the super-user forcibly invalidates all
301 existing performance counter context. This can in turn cause all invocations of
302 the \fBcputrack\fR command, and other users of performance counter context, to
303 exit prematurely with unspecified errors.
304 .sp
305 .LP
306 If \fBcpustat\fR is invoked on a system that has \fBCPU\fR performance counters
307 which are not supported by Solaris, the following message appears:
308 .sp
309 .in +2
310 .nf
311 cpustat: cannot access performance counters - Operation not applicable
312 .fi
313 .in -2
314 .sp

316 .sp
317 .LP
318 This error message implies that \fBcpc_open()\fR has failed and is documented
319 in \fBcpc_open\fR(3CPC). Review this documentation for more information about
320 the problem and possible solutions.
321 .sp
322 .LP
323 If a short interval is requested, \fBcpustat\fR might not be able to keep up
324 with the desired sample rate. In this case, some samples might be dropped.

```

```

325 .SH ATTRIBUTES
326 .sp
327 .LP
328 See \fBattributes\fR(5) for descriptions of the following attributes:
329 .sp

331 .sp
332 .TS
333 box;
334 c | c
335 l | l .
336 ATTRIBUTE TYPE ATTRIBUTE VALUE
337 -
338 Interface Stability Evolving
339 .TE

341 .SH SEE ALSO
342 .sp
343 .LP
344 \fBcputrack\fR(1), \fBbawk\fR(1), \fBperl\fR(1), \fBbiostat\fR(1M),
344 \fBcputrack\fR(1), \fBbnawk\fR(1), \fBperl\fR(1), \fBbiostat\fR(1M),
345 \fBprstat\fR(1M), \fBpsrset\fR(1M), \fBvmstat\fR(1M), \fBcpc\fR(3CPC),
346 \fBcpc_open\fR(3CPC), \fBcpc_bind_cpu\fR(3CPC), \fBgethrtime\fR(3C),
347 \fBstrtoll\fR(3C), \fBlibcpc\fR(3LIB), \fBattributes\fR(5)
348 .SH NOTES
349 .sp
350 .LP
351 When \fBcpustat\fR is run on a Pentium 4 with HyperThreading enabled, a CPC set
352 is bound to only one logical CPU of each physical CPU. See
353 \fBcpc_bind_cpu\fR(3CPC).

```

7065 Sun May 26 18:23:37 2013

new/usr/src/man/man5/largefile.5

XXX Remove nawk(1)

```

1  \' te
2  .\' Copyright (c) 2007 Sun Microsystems, Inc. All Rights Reserved
3  .\' Portions Copyright (c) 1982-2007 AT&T Knowledge Ventures
4  .\' The contents of this file are subject to the terms of the Common Development
5  .\' or http://www.opensolaris.org/os/licensing. See the License for the specifi
6  .\' the following below this CDDL HEADER, with the fields enclosed by brackets "
7  .TH LARGEFILE 5 "Nov 2, 2007"
8  .SH NAME
9  largefile \- large file status of utilities
10 .SH DESCRIPTION
11 .sp
12 .LP
13 A \fIlarge file\&R is a regular file whose size is greater than or equal to 2
14 Gbyte ( 2^31 bytes). A \fIsmall file\&R is a regular file whose size is less
15 than 2 Gbyte.
16 .SS "Large file aware utilities"
17 .sp
18 .LP
19 A utility is called \fIlarge file aware\&R if it can process large files in the
20 same manner as it does small files. A utility that is large file aware is able
21 to handle large files as input and generate as output large files that are
22 being processed. The exception is where additional files are used as system
23 configuration files or support files that can augment the processing. For
24 example, the \fBfile\&R utility supports the \fB-m\&R option for an alternative
25 "magic" file and the \fB-f\&R option for a support file that can contain a list
26 of file names. It is unspecified whether a utility that is large file aware
27 will accept configuration or support files that are large files. If a large
28 file aware utility does not accept configuration or support files that are
29 large files, it will cause no data loss or corruption upon encountering such
30 files and will return an appropriate error.
31 .sp
32 .LP
33 The following \fB/usr/bin\&R utilities are large file aware:
34 .sp

36 .sp
37 .TS
38 l l l l l
39 l l l l l .
40 \fBbadb\&R \fBaliasadm\&R \fBbawk\&R \fBbddiff\&R \fBbcats\&R
41 \fBbchgrp\&R \fBbchmod\&R \fBbchown\&R \fBbcksum\&R \fBbcmp\&R
42 \fBbcompress\&R \fBbcp\&R \fBbcsh\&R \fBbcsplit\&R \fBbcut\&R
43 \fBbdd\&R \fBdircmp\&R \fBbdu\&R \fBbegrep\&R \fBbfgrep\&R
44 \fBbfile\&R \fBbfind\&R \fBbftp\&R \fBbgetconf\&R \fBbgrep\&R
45 \fBbgzip\&R \fBbhead\&R \fBbjoin\&R \fBbjsh\&R \fBbksh\&R
46 \fBbksh93\&R \fBbln\&R \fBbls\&R \fBbmailcompat\&R \fBbmails
47 \fBbmdb\&R \fBbmdir\&R \fBbmkfifo\&R \fBbmore\&R \fBbmv\&R
48 \fBbpage\&R \fBbpaste\&R \fBbpathchk\&R \fBbpg\&R
48 \fBbnawk\&R \fBbpage\&R \fBbpaste\&R \fBbpathchk\&R \fBbpg\&R
49 \fBbpraliases\&R \fBbrcp\&R \fBbremsh\&R \fBbrksh\&R \fBbrksh93\&R
50 \fBbrm\&R \fBbrmdir\&R \fBbrsh\&R \fBbsed\&R \fBbsh\&R
51 \fBbsort\&R \fBbsplit\&R \fBbsum\&R \fBbtail\&R \fBbtar\&R
52 \fBbtee\&R \fBbtest\&R \fBbtouch\&R \fBbtr\&R \fBbuncompress\&R
53 \fBbuudcode\&R \fBbuencode\&R \fBbvacation\&R \fBbwc\&R \fBbzcat\&R
54 .TE

56 .sp
57 .LP
58 The following \fB/usr/xpg4/bin\&R utilities are large file aware:
59 .sp

```

```

61 .sp
62 .TS
63 l l l l l
64 l l l l l .
65 \fBbawk\&R \fBbcp\&R \fBbchgrp\&R \fBbchown\&R \fBbdu\&R
66 \fBbegrep\&R \fBbfgrep\&R \fBbfile\&R \fBbgrep\&R \fBbln\&R
67 \fBbls\&R \fBbmore\&R \fBbmv\&R \fBbrm\&R \fBbsed\&R
68 \fBbsh\&R sort tail tr
69 .TE

71 .sp
72 .LP
73 The following \fB/usr/xpg6/bin\&R utilities are large file aware:
74 .sp

76 .sp
77 .TS
78 l l l l l
79 \fBbgetconf\&R \fBbls\&R \fBbtr\&R
80 .TE

82 .sp
83 .LP
84 The following \fB/usr/sbin\&R utilities are large file aware:
85 .sp

87 .sp
88 .TS
89 l l l l l
90 l l l l l .
91 \fBbeditmap\&R \fBbininstall\&R \fBbmake\&R \fBbmkfile\&R \fBbmknod\&R
92 \fBbmvdir\&R \fBbswap\&R
93 .TE

95 .sp
96 .LP
97 The following \fB/usr/lib\&R utilities are large file aware:
98 .sp

100 .sp
101 .TS
102 l l l l l
103 \fBbmail.local\&R \fBbsendmail\&R \fBbsmrsh\&R
104 .TE

106 .sp
107 .LP
108 See the USAGE section of the \fBbswap\&R(1M) manual page for limitations of
109 \fBbswap\&R on block devices greater than 2 Gbyte on a 32-bit operating system.
110 .sp
111 .LP
112 The following \fB/usr/ucb\&R utilities are large file aware:
113 .sp

115 .sp
116 .TS
117 l l l l l
118 l l l l l .
119 \fBbchown\&R \fBbfrom\&R \fBbln\&R \fBbls\&R \fBbsed\&R
120 \fBbsum\&R \fBbtouch\&R
121 .TE

123 .sp
124 .LP
125 The \fB/usr/bin/cpio\&R and \fB/usr/bin/pax\&R utilities are large file aware,
126 but cannot archive a file whose size exceeds 8 Gbyte - 1 byte.

```

```

127 .sp
128 .LP
129 The \fB/usr/bin/truss\fR utilities has been modified to read a dump file and
130 display information relevant to large files, such as offsets.
131 .SS "cachefs file systems"
132 .sp
133 .LP
134 The following \fB/usr/bin\fR utilities are large file aware for \fBcachefs\fR
135 file systems:
136 .sp

138 .sp
139 .TS
140 l l l l .
141 \fBcachefspack\fR          \fBcachefsstat\fR
142 .TE

144 .sp
145 .LP
146 The following \fB/usr/sbin\fR utilities are large file aware for \fBcachefs\fR
147 file systems:
148 .sp

150 .sp
151 .TS
152 l l l l
153 l l l l .
154 \fBcachefslog\fR          \fBcachefswsize\fR      \fBcfsadmin\fR  \fBfscck\fR
155 \fBmount\fR              \fBumount\fR
156 .TE

158 .SS "nfs file systems"
159 .sp
160 .LP
161 The following utilities are large file aware for \fBnfs\fR file systems:
162 .sp

164 .sp
165 .TS
166 l l
167 l l .
168 \fB/usr/lib/autofs/automountd\fR      \fB/usr/sbin/mount\fR
169 \fB/usr/lib/nfs/rquotad\fR
170 .TE

172 .SS "ufs file systems"
173 .sp
174 .LP
175 The following \fB/usr/bin\fR utility is large file aware for \fBbufs\fR file
176 systems:
177 .sp
178 .LP
179 \fBbdf\fR
180 .sp
181 .LP
182 The following \fB/usr/lib/nfs\fR utility is large file aware for \fBbufs\fR file
183 systems:
184 .sp
185 .LP
186 \fBbrquotad\fR
187 .sp
188 .LP
189 The following \fB/usr/xpg4/bin\fR utility is large file aware for \fBbufs\fR
190 file systems:
191 .sp
192 .LP

```

```

193 \fBbdf\fR
194 .sp
195 .LP
196 The following \fB/usr/sbin\fR utilities are large file aware for \fBbufs\fR file
197 systems:
198 .sp

200 .sp
201 .TS
202 l l l l l
203 l l l l l .
204 \fBclri\fR          \fBdcopy\fR      \fBedquota\fR      \fBff\fR          \fBfscck\fR
205 \fBfsdb\fR          \fBfsirand\fR    \fBfstyp\fR      \fBlabelit\fR    \fBblockfs\fR
206 \fBmkfs\fR          \fBmount\fR      \fBncheck\fR    \fBnewfs\fR      \fBquot\fR
207 \fBquota\fR          \fBquotacheck\fR \fBquotaoff\fR  \fBquotaon\fR    \fBrepqu
208 \fBtunefs\fR        \fBufsdump\fR   \fBufsrestore\fR \fBumount\fR
209 .TE

211 .SS "Large file safe utilities"
212 .sp
213 .LP
214 A utility is called \fBlarge file safe\fR if it causes no data loss or
215 corruption when it encounters a large file. A utility that is large file safe
216 is unable to process properly a large file, but returns an appropriate error.
217 .sp
218 .LP
219 The following \fB/usr/bin\fR utilities are large file safe:
220 .sp

222 .sp
223 .TS
224 l l l l l
225 l l l l l .
226 \fBaudioconvert\fR    \fBaudioplay\fR  \fBaudiorecord\fR    \fBcomm\fR
227 \fBdiff3\fR          \fBdiffmk\fR     \fBed\fR           \fBlp\fR         \fBmail\fR
228 \fBmailcompat\fR     \fBmailstats\fR \fBmailx\fR         \fBpack\fR        \fBpcat\fR
229 \fBred\fR            \fBrmail\fR      \fBsdiff\fR        \fBunpack\fR      \fBvi\fR
230 \fBview\fR
231 .TE

233 .sp
234 .LP
235 The following \fB/usr/xpg4/bin\fR utilities are large file safe:
236 .sp

238 .sp
239 .TS
240 l l l l l
241 \fBed\fR            \fBvi\fR          \fBview\fR
242 .TE

244 .sp
245 .LP
246 The following \fB/usr/xpg6/bin\fR utility is large file safe:
247 .sp

249 .sp
250 .TS
251 l l l l l
252 \fBed\fR
253 .TE

255 .sp
256 .LP
257 The following \fB/usr/sbin\fR utilities are large file safe:
258 .sp

```

```
260 .sp
261 .TS
262 l l l l l .
263 lpfilter          lpforms
264 .TE

266 .sp
267 .LP
268 The following \fB/usr/ucb\fR utilities are large file safe:
269 .sp

271 .sp
272 .TS
273 l l l l l .
274 \fBMail\fR      \fBldr\fR
275 .TE

277 .SH SEE ALSO
278 .sp
279 .LP
280 \fBblk64\fR(5), \fBblkcompile\fR(5), \fBblkcompile64\fR(5)
```

```

*****
24833 Sun May 26 18:23:39 2013
new/usr/src/pkg/Makefile
XXX Remove awk(1)
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #

22 #
23 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
24 #

26 include $(SRC)/Makefile.master
27 include $(SRC)/Makefile.buildnum

29 #
30 # Make sure we're getting a consistent execution environment for the
31 # embedded scripts.
32 #
33 SHELL= /usr/bin/ksh93

35 #
36 # To suppress package dependency generation on any system, regardless
37 # of how it was installed, set SUPPRESSPKGDEP=true in the build
38 # environment.
39 #
40 SUPPRESSPKGDEP= false

42 #
43 # Comment this line out or set "PKGDEBUG=" in your build environment
44 # to get more verbose output from the make processes in usr/src/pkg
45 #
46 PKGDEBUG= @

48 #
49 # Cross platform packaging notes
50 #
51 # By default, we package the proto area from the same architecture as
52 # the packaging build. In other words, if you're running nightly or
53 # bldenv on an x86 platform, it will take objects from the x86 proto
54 # area and use them to create x86 repositories.
55 #
56 # If you want to create repositories for an architecture that's
57 # different from $(uname -p), you do so by setting PKGMACH in your
58 # build environment.
59 #
60 # For this to work correctly, the following must all happen:
61 #

```

```

62 # 1. You need the desired proto area, which you can get either by
63 # doing a gatekeeper-style build with the -U option to
64 # nightly(1), or by using rsync. If you don't do this, you will
65 # get packaging failures building all packages, because pkgsend
66 # is unable to find the required binaries.
67 # 2. You need the desired tools proto area, which you can get in the
68 # same ways as the normal proto area. If you don't do this, you
69 # will get packaging failures building onbld, because pkgsend is
70 # unable to find the tools binaries.
71 # 3. The remainder of this Makefile should never refer directly to
72 # $(MACH). Instead, $(PKGMACH) should be used whenever an
73 # architecture-specific path or token is needed. If this is done
74 # incorrectly, then packaging will fail, and you will see the
75 # value of $(uname -p) instead of the value of $(PKGMACH) in the
76 # commands that fail.
77 # 4. Each time a rule in this Makefile invokes $(MAKE), it should
78 # pass PKGMACH=$(PKGMACH) explicitly on the command line. If
79 # this is done incorrectly, then packaging will fail, and you
80 # will see the value of $(uname -p) instead of the value of
81 # $(PKGMACH) in the commands that fail.
82 #
83 # Refer also to the convenience targets defined later in this
84 # Makefile.
85 #
86 PKGMACH= $(MACH)

88 #
89 # ROOT, TOOLS_PROTO, and PKGARCHIVE should be set by nightly or
90 # bldenv. These macros translate them into terms of $PKGMACH, instead
91 # of $ARCH.
92 #
93 PKGROOT.cmd= print $(ROOT) | sed -e s:/root_$(MACH):/root_$(PKGMACH):
94 PKGROOT= $(PKGROOT.cmd:sh)
95 TOOLSROOT.cmd= print $(TOOLS_PROTO) | sed -e s:/root_$(MACH):/root_$(PKGMACH):
96 TOOLSROOT= $(TOOLSROOT.cmd:sh)
97 PKGDEST.cmd= print $(PKGARCHIVE) | sed -e s:/$(MACH):/$(PKGMACH):/
98 PKGDEST= $(PKGDEST.cmd:sh)

100 EXCEPTIONS= packaging

102 PKGMOGRIFY= pkgmogrify

104 #
105 # Always build the redistributable repository, but only build the
106 # nonredistributable bits if we have access to closed source.
107 #
108 # Some objects that result from the closed build are still
109 # redistributable, and should be packaged as part of an open-only
110 # build. Access to those objects is provided via the closed-bins
111 # tarball. See usr/src/tools/scripts/bindrop.sh for details.
112 #
113 REPOS= redist

115 #
116 # The packages directory will contain the processed manifests as
117 # direct build targets and subdirectories for package metadata extracted
118 # incidentally during manifest processing.
119 #
120 # Nothing underneath $(PDIR) should ever be managed by SCM.
121 #
122 PDIR= packages.$(PKGMACH)

124 #
125 # The tools proto must be specified for dependency generation.
126 # Publication from the tools proto area is managed in the
127 # publication rule.

```



```

128 #
129 $(PDIR)/developer-build-onbld.dep:= PKGROOT= $(TOOLSROOT)

131 PKGPUBLISHER= $(PKGPUBLISHER_REDIST)

133 #
134 # To get these defaults, manifests should simply refer to $(PKGVERS).
135 #
136 PKGVERS_COMPONENT= 0.$(RELEASE)
137 PKGVERS_BUILTON= $(RELEASE)
138 PKGVERS_BRANCH= 0.$(ONNV_BUILDNUM)
139 PKGVERS= $(PKGVERS_COMPONENT),$(PKGVERS_BUILTON)-$(PKGVERS_BRANCH)

141 #
142 # The ARCH32 and ARCH64 macros are used in the manifests to express
143 # architecture-specific subdirectories in the installation paths
144 # for isaexec'd commands.
145 #
146 # We can't simply use $(MACH32) and $(MACH64) here, because they're
147 # only defined for the build architecture. To do cross-platform
148 # packaging, we need both values.
149 #
150 i386_ARCH32= i86
151 sparc_ARCH32= sparcv7
152 i386_ARCH64= amd64
153 sparc_ARCH64= sparcv9

155 OPENSSL = /usr/bin/openssl
156 OPENSSL10.cmd = $(OPENSSL) version | $(AWK) '{if($$2<1){print "\043";}}'
157 OPENSSL10_ONLY = $(OPENSSL) version | $(NAWK) '{if($$2<1){print "\043";}}'
158 OPENSSL10_ONLY = $(OPENSSL10.cmd:sh)

159 #
160 # macros and transforms needed by pkgmogrify
161 #
162 # If you append to this list using target-specific assignments (:=),
163 # be very careful that the targets are of the form $(PDIR)/pkgname. If
164 # you use a higher level target, or a package list, you'll trigger a
165 # complete reprocessing of all manifests because they'll fail command
166 # dependency checking.
167 #
168 PM_TRANSFORMS= common_actions publish restart_fmri facets defaults \
169 extract_metadata
170 PM_INC= transforms manifests

172 PKGMOG_DEFINES= \
173 i386_ONLY=$(POUND_SIGN) \
174 sparc_ONLY=$(POUND_SIGN) \
175 OPENSSL10_ONLY=$(OPENSSL10_ONLY) \
176 $(PKGARCH)_ONLY= \
177 ARCH=$(PKGARCH) \
178 ARCH32=$( $(PKGARCH)_ARCH32) \
179 ARCH64=$( $(PKGARCH)_ARCH64) \
180 PKGVERS_COMPONENT=$(PKGVERS_COMPONENT) \
181 PKGVERS_BUILTON=$(PKGVERS_BUILTON) \
182 PKGVERS_BRANCH=$(PKGVERS_BRANCH) \
183 PKGVERS=$(PKGVERS)

185 PKGDEP_TOKENS_i386= \
186 'PLATFORM=i86hvm' \
187 'PLATFORM=i86pc' \
188 'PLATFORM=i86xpv' \
189 'ISALIST=amd64' \
190 'ISALIST=i386'
191 PKGDEP_TOKENS_sparc= \
192 'PLATFORM=sun4u' \

```

```

193 'PLATFORM=sun4v' \
194 'ISALIST=sparcv9' \
195 'ISALIST=sparc'
196 PKGDEP_TOKENS= $(PKGDEP_TOKENS_$(PKGARCH))

198 #
199 # The package lists are generated with $(PKGDEP_TYPE) as their
200 # dependency types, so that they can be included by either an
201 # incorporation or a group package.
202 #
203 $(PDIR)/osnet-redist.mog := PKGDEP_TYPE= require
204 $(PDIR)/osnet-incorporation.mog:= PKGDEP_TYPE= incorporate

206 PKGDEP_INCORP= \
207 depend_fmri=consolidation/osnet/osnet-incorporation type=require

209 #
210 # All packaging build products should go into $(PDIR), so they don't
211 # need to be included separately in CLOBBERFILES.
212 #
213 CLOBBERFILES= $(PDIR) proto_list_$(PKGARCH)

215 #
216 # By default, PKGS will list all manifests. To build and/or publish a
217 # subset of packages, override this on the command line or in the
218 # build environment and then reference (implicitly or explicitly) the all
219 # or install targets.
220 #
221 MANIFESTS :sh= (cd manifests; print *.mf)
222 PKGS= $(MANIFESTS:%.mf=)
223 DEP_PKGS= $(PKGS:%=$(PDIR)/%.dep)
224 PROC_PKGS= $(PKGS:%=$(PDIR)/%.mog)

226 #
227 # Track the synthetic manifests separately so we can properly express
228 # build rules and dependencies. The synthetic and real packages use
229 # different sets of transforms and macros for pkgmogrify.
230 #
231 SYNTH_PKGS= osnet-incorporation osnet-redist
232 DEP_SYNTH_PKGS= $(SYNTH_PKGS:%=$(PDIR)/%.dep)
233 PROC_SYNTH_PKGS= $(SYNTH_PKGS:%=$(PDIR)/%.mog)

235 #
236 # Root of pkg image to use for dependency resolution
237 # Normally / on the machine used to build the binaries
238 #
239 PKGDEP_RESOLVE_IMAGE = /

241 #
242 # For each package, we determine the target repository based on
243 # manifest-embedded metadata. Because we make that determination on
244 # the fly, the publication target cannot be expressed as a
245 # subdirectory inside the unknown-by-the-makefile target repository.
246 #
247 # In order to limit the target set to real files in known locations,
248 # we use a ".pub" file in $(PDIR) for each processed manifest, regardless
249 # of content or target repository.
250 #
251 PUB_PKGS= $(SYNTH_PKGS:%=$(PDIR)/%.pub) $(PKGS:%=$(PDIR)/%.pub)

253 #
254 # Any given repository- and status-specific package list may be empty,
255 # but we can only determine that dynamically, so we always generate all
256 # lists for each repository we're building.
257 #
258 # The meanings of each package status are as follows:

```

```

259 #
260 #     PKGSTAT      meaning
261 #     -----
262 #     noincorp     Do not include in incorporation or group package
263 #     obsolete     Include in incorporation, but not group package
264 #     renamed      Include in incorporation, but not group package
265 #     current      Include in incorporation and group package
266 #
267 # Since the semantics of the "noincorp" package status dictate that
268 # such packages are not included in the incorporation or group packages,
269 # there is no need to build noincorp package lists.
270 #
271 PKGLISTS= \
272   $(REPOS:=$(PDIR)/packages.%.current) \
273   $(REPOS:=$(PDIR)/packages.%.renamed) \
274   $(REPOS:=$(PDIR)/packages.%.obsolete)
275 #
276 .KEEP_STATE:
277 #
278 .PARALLEL: $(PKGS) $(PROC_PKGS) $(DEP_PKGS) \
279   $(PROC_SYNTH_PKGS) $(DEP_SYNTH_PKGS) $(PUB_PKGS)
280 #
281 #
282 # For a single manifest, the dependency chain looks like this:
283 #
284 #     raw manifest (mypkg.mf)
285 #     |
286 #     | use pkgmogrify to process raw manifest
287 #     |
288 #     processed manifest (mypkg.mog)
289 #     |
290 #     | * use pkgdepend generate to generate dependencies
291 #     |
292 #     manifest with TBD dependencies (mypkg.dep)
293 #     |
294 #     | % use pkgdepend resolve to resolve dependencies
295 #     |
296 #     manifest with dependencies resolved (mypkg.res)
297 #     |
298 #     | use pkgsend to publish the package
299 #     |
300 #     placeholder to indicate successful publication (mypkg.pub)
301 #
302 # * This may be suppressed via SUPPRESSPKGDEP. The resulting
303 # packages will install correctly, but care must be taken to
304 # install all dependencies, because pkg will not have the input
305 # it needs to determine this automatically.
306 #
307 # % This is included in this diagram to make the picture complete, but
308 # this is a point of synchronization in the build process.
309 # Dependency resolution is actually done once on the entire set of
310 # manifests, not on a per-package basis.
311 #
312 # The full dependency chain for generating everything that needs to be
313 # published, without actually publishing it, looks like this:
314 #
315 #     processed synthetic packages
316 #     |
317 #     | package lists
318 #     |
319 #     | processed real packages
320 #     |
321 #     | package dir
322 #     |
323 #     | real package manifests
324 #
325 # Here, each item is a set of real or synthetic packages. For this
326 # portion of the build, no reference is made to the proto area. It is

```

```

325 # therefore suitable for the "all" target, as opposed to "install."
326 #
327 # Since each of these steps is expressed explicitly, "all" need only
328 # depend on the head of the chain.
329 #
330 # From the end of manifest processing, the publication dependency
331 # chain looks like this:
332 #
333 #     repository metadata (catalogs and search indices)
334 #     |
335 #     | pkg.depotd
336 #     |
337 #     | published packages
338 #     |
339 #     | pkgsend publish
340 #     |
341 #     | repositories
342 #     |
343 #     | resolved dependencies
344 #     |
345 #     | pkgdepend resolve
346 #     |
347 #     | generated dependencies
348 #     |
349 #     | pkgdepend
350 #     |
351 #     | processed manifests
352 #
353 #
354 ALL_TARGETS= $(PROC_SYNTH_PKGS) proto_list_$(PKGSMACH)
355 #
356 all: $(ALL_TARGETS)
357 #
358 # This will build the directory to contain the processed manifests
359 # and the metadata symlinks.
360 #
361 $(PDIR):
362   @print "Creating $(@)"
363   $(PKGDEBUG)$(INS.dir)
364 #
365 # This rule resolves dependencies across all published manifests.
366 #
367 # We shouldn't have to ignore the error from pkgdepend, but until
368 # 16012 and its dependencies are resolved, pkgdepend will always exit
369 # with an error.
370 #
371 $(PDIR)/gendeps: $(DEP_SYNTH_PKGS) $(DEP_PKGS)
372   -$(PKGDEBUG)if [ "$(SUPPRESSPKGDEP)" = "true" ]; then \
373     print "Suppressing dependency resolution"; \
374     for p in $(DEP_PKGS:%.dep=%); do \
375       $(CP) $$p.dep $$p.res; \
376     done; \
377   else \
378     print "Resolving dependencies"; \
379     pkgdepend -R $(PKGDEP_RESOLVE_IMAGE) resolve \
380       -m $(DEP_SYNTH_PKGS) $(DEP_PKGS); \
381     for p in $(DEP_SYNTH_PKGS:%.dep=%) $(DEP_PKGS:%.dep=%); do \
382       if [ "$$(print $$p.metadata.*)" = \
383         "$$(print $$p.metadata.noincorp.*)" ]; \
384     then \
385       print "Removing dependency versions from $$p"; \
386       $(PKMGRIFY) $(PKGMOG_VERBOSE) \
387         -O $$p.res -I transforms \
388         strip_versions $$p.dep.res; \
389     else \
390       $(RM) $$p.dep.res; \

```

```

391             $(MV) $$p.dep.res $$p.res; \
392             fi; \
393         done; \
394     fi
395     $(PKGDEBUG)$ (TOUCH) $(@)

397 install: $(ALL_TARGETS) repository-metadata

399 repository-metadata: publish_pkgs
400     @print "Creating repository metadata"
401     $(PKGDEBUG)for r in $(REPOS); do \
402         /usr/lib/pkg.depotd -d $(PKGDEST)/repo.$$r \
403         --add-content --exit-ready; \
404     done

406 #
407 # Since we create zero-length processed manifests for a graceful abort
408 # from pkgmogrify, we need to detect that here and make no effort to
409 # publish the package.
410 #
411 # For all other packages, we publish them regardless of status. We
412 # derive the target repository as a component of the metadata-derived
413 # symlink for each package.
414 #
415 publish_pkgs: $(REPOS:%=$(PKGDEST)/repo.%) $(PDIR)/gendeps .WAIT $(PUB_PKGS)

417 #
418 # Before publishing, we want to pull the license files from $CODEMGR_WS
419 # into the proto area. This allows us to NOT pass $SRC (or
420 # $CODEMGR_WS) as a basedir for publication.
421 #
422 $(PUB_PKGS): stage-licenses

424 #
425 # Initialize the empty on-disk repositories
426 #
427 $(REPOS:%=$(PKGDEST)/repo.%):
428     @print "Initializing $(@F)"
429     $(PKGDEBUG)$ (INS.dir)
430     $(PKGDEBUG)pkgsend -s file://$(@) create-repository \
431     --set-property publisher.prefix=$(PKGPUBLISHER)

433 #
434 # rule to process real manifests
435 #
436 # To allow redistributability and package status to change, we must
437 # remove not only the actual build target (the processed manifest), but
438 # also the incidental ones (the metadata-derived symlinks).
439 #
440 # If pkgmogrify exits cleanly but fails to create the specified output
441 # file, it means that it encountered an abort directive. That means
442 # that this package should not be published for this particular build
443 # environment. Since we can't prune such packages from $(PKGS)
444 # retroactively, we need to create an empty target file to keep make
445 # from trying to rebuild it every time. For these empty targets, we
446 # do not create metadata symlinks.
447 #
448 # Automatic dependency resolution to files is also done at this phase of
449 # processing. The skipped packages are skipped due to existing bugs
450 # in pkgdepend.
451 #
452 # The incorporation dependency is tricky: it needs to go into all
453 # current and renamed manifests (ie all incorporated packages), but we
454 # don't know which those are until after we run pkgmogrify. So
455 # instead of expressing it as a transform, we tack it on ex post facto.
456 #

```

```

457 # Implementation notes:
458 #
459 # - The first $(RM) must not match other manifests, or we'll run into
460 # race conditions with parallel manifest processing.
461 #
462 # - The make macros [ie $(MACRO)] are evaluated when the makefile is
463 # read in, and will result in a fixed, macro-expanded rule for each
464 # target enumerated in $(PROC_PKGS).
465 #
466 # - The shell variables (ie $$VAR) are assigned on the fly, as the rule
467 # is executed. The results may only be referenced in the shell in
468 # which they are assigned, so from the perspective of make, all code
469 # that needs these variables needs to be part of the same line of
470 # code. Hence the use of command separators and line continuation
471 # characters.
472 #
473 # - The extract_metadata transforms are designed to spit out shell
474 # variable assignments to stdout. Those are published to the
475 # .vars temporary files, and then used as input to the eval
476 # statement. This is done in stages specifically so that pkgmogrify
477 # can signal failure if the manifest has a syntactic or other error.
478 # The eval statement should begin with the default values, and the
479 # output from pkgmogrify (if any) should be in the form of a
480 # variable assignment to override those defaults.
481 #
482 # - When this rule completes execution, it must leave an updated
483 # target file ($@) in place, or make will reprocess the package
484 # every time it encounters it as a dependency. Hence the "touch"
485 # statement to ensure that the target is created, even when
486 # pkgmogrify encounters an abort in the publish transforms.
487 #

489 .SUFFIXES: .mf .mog .dep .res .pub

491 $(PDIR)/%.mog: manifests/%.mf
492     @print "Processing manifest $(<F)"
493     @env PKGFMT_OUTPUT=v1 pkgfmt -c $<
494     $(PKGDEBUG)$ (RM) $(@) $(@:%.mog=%) $(@:%.mog=%.nodepend) \
495     $(@:%.mog=%.lics) $(PDIR)/$(@F:%.mog=%).metadata.* $(@).vars
496     $(PKGDEBUG)$ (PKGMOGRIFY) $(PKGMOG_VERBOSE) $(PM_INC:%=-I %) \
497     $(PKGMOG_DEFINES:%=-D %) -P $(@).vars -O $(@) \
498     $(<) $(PM_TRANSFORMS)
499     $(PKGDEBUG)eval REPO=redist PKGSTAT=current NODEPEND=$(SUPPRESSPKGDEP) \
500     `$(CAT) -s $(@).vars`; \
501     if [ -f $(@) ]; then \
502         if [ "$$NODEPEND" != "false" ]; then \
503             $(TOUCH) $(@:%.mog=%.nodepend); \
504         fi; \
505         $(LN) -s $(@F) \
506         $(PDIR)/$(@F:%.mog=%).metadata.$$PKGSTAT.$$REPO; \
507         if [ \ ( "$$PKGSTAT" = "current" \ ) -o \
508             \ ( "$$PKGSTAT" = "renamed" \ ) ]; \
509             then print $(PKGDEP_INCORP) >> $(@); \
510         fi; \
511         print $$LICS > $(@:%.mog=%.lics); \
512     else \
513         $(TOUCH) $(@) $(@:%.mog=%.lics); \
514     fi
515     $(PKGDEBUG)$ (RM) $(@).vars

517 $(PDIR)/%.dep: $(PDIR)/%.mog
518     @print "Generating dependencies for $(<F)"
519     $(PKGDEBUG)$ (RM) $(@)
520     $(PKGDEBUG)if [ ! -f $(@:%.dep=%.nodepend) ]; then \
521         pkgdepend generate -m $(PKGDEP_TOKENS:%=-D %) $(<) \
522         $(PKGROOT) > $(@); \

```

```

523     else \
524         $(CP) $(<) $(@); \
525     fi

527 #
528 # The full chain implies that there should be a .dep.res suffix rule,
529 # but dependency generation is done on a set of manifests, rather than
530 # on a per-manifest basis.  Instead, see the gendeps rule above.
531 #

533 $(PDIR)/%.pub: $(PDIR)/%.res
534     $(PKGDEBUG)m=$(basename $(@:%.pub=%).metadata.*); \
535     r=${m#$(@F:%.pub=%).metadata.+(?).}; \
536     if [ -s $(<) ]; then \
537         print "Publishing $(@F:%.pub=) to $$r repository"; \
538         pkgsend -s file://$(PKGDEST)/repo.$$r publish \
539             -d $(PKGROOT) -d $(TOOLSROOT) \
540             -d license_files -d $(PKGROOT)/licenses \
541             --fmri-in-manifest --no-index --no-catalog $(<) \
542             > /dev/null; \
543     fi; \
544     $(TOUCH) $(@);

546 #
547 # rule to build the synthetic manifests
548 #
549 # This rule necessarily has PKGDEP_TYPE that changes according to
550 # the specific synthetic manifest.  Rather than escape command
551 # dependency checking for the real manifest processing, or failing to
552 # express the (indirect) dependency of synthetic manifests on real
553 # manifests, we simply split this rule out from the one above.
554 #
555 # The implementation notes from the previous rule are applicable
556 # here, too.
557 #
558 $(PROC_SYNTH_PKGS): $(PKGLISTS) ${@F:%.mog=%.mf}
559     @print "Processing synthetic manifest $(@F:%.mog=%.mf)"
560     $(PKGDEBUG)$ (RM) $(@) $(PDIR)/${@F:%.mog=%}.metadata.* $(@).vars
561     $(PKGDEBUG)$ (PKGMOGRIFY) $(PKGMOG_VERBOSE) -I transforms -I $(PDIR) \
562     $(PKGMOG_DEFINES:%=-D %) -D PKGDEP_TYPE=$(PKGDEP_TYPE) \
563     -P $(@).vars -O $(@) $(@F:%.mog=%.mf) \
564     $(PM_TRANSFORMS) synthetic
565     $(PKGDEBUG)eval REPO=redist PKGSTAT=current `$(CAT) -s $(@).vars`; \
566     if [ -f $(@) ]; then \
567         $(LN) -s $(@F) \
568             $(PDIR)/${@F:%.mog=%}.metadata.$$PKGSTAT.$$REPO; \
569     else \
570         $(TOUCH) $(@); \
571     fi
572     $(PKGDEBUG)$ (RM) $(@).vars

574 $(DEP_SYNTH_PKGS): ${@:%.dep=%.mog}
575     @print "Skipping dependency generation for $(@F:%.dep=%)"
576     $(PKGDEBUG)$ (CP) $(@:%.dep=%.mog) $(@)

578 clean:

580 clobber: clean
581     $(RM) -r $(CLOBBERFILES)

583 #
584 # This rule assumes that all links in the $PKGSTAT directories
585 # point to valid manifests, and will fail the make run if one
586 # does not contain an fmri.
587 #
588 # We do this in the BEGIN action instead of using pattern matching

```

```

589 # because we expect the fmri to be at or near the first line of each input
590 # file, and this way lets us avoid reading the rest of the file after we
591 # find what we need.
592 #
593 # We keep track of a failure to locate an fmri, so we can fail the
594 # make run, but we still attempt to process each package in the
595 # repo/pkgstat-specific subdir, in hopes of maybe giving some
596 # additional useful info.
597 #
598 # The protolist is used for bfu archive creation, which may be invoked
599 # interactively by the user.  Both protolist and PKGLISTS targets
600 # depend on $(PROC_PKGS), but protolist builds them recursively.
601 # To avoid collisions, we insert protolist into the dependency chain
602 # here.  This has two somewhat subtle benefits: it allows bfu archive
603 # creation to work correctly, even when -a was not part of NIGHTLY_OPTIONS,
604 # and it ensures that a protolist file here will always correspond to the
605 # contents of the processed manifests, which can vary depending on build
606 # environment.
607 #
608 $(PKGLISTS): $(PROC_PKGS)
609     $(PKGDEBUG)sdotr=${@F:packages.%=}; \
610     r=${sdotr%+(?)}; s=${sdotr#+(?)}; \
611     print "Generating $$r $$s package list"; \
612     $(RM) $(@); $(TOUCH) $(@); \
613     $(AWK) 'BEGIN { \
614         $(NAWK) 'BEGIN { \
615             if (ARGC < 2) { \
616                 exit; \
617             } \
618             retcode = 0; \
619             for (i = 1; i < ARGC; i++) { \
620                 do { \
621                     e = getline f < ARGV[i]; \
622                     } while ((e == 1) && (f !~ /name=pkg.fmri/)); \
623                     close(ARGV[i]); \
624                     if (e == 1) { \
625                         l = split(f, a, "="); \
626                         print "depend fmri=" a[1], \
627                             "type=$(PKGDEP_TYPE)"; \
628                     } else { \
629                         print "no fmri in " ARGV[i] >> "/dev/stderr"; \
630                         retcode = 2; \
631                     } \
632                 } \
633             }' `find $(PDIR) -type l -a \( $(PKGS:%=-name %.metadata.$$s.$$r -o) \
634                 -name NOSUCHFILE \)` >> $(@)

636 #
637 # rules to validate proto area against manifests, check for safe
638 # file permission modes, and generate a faux proto list
639 #
640 # For the check targets, the dependencies on $(PROC_PKGS) is specified
641 # as a subordinate make process in order to suppress output.
642 #
643 makesilent:
644     @$(MAKE) -e $(PROC_PKGS) PKGMACH=$(PKGMACH) \
645         SUPPRESSPKGDEP=$(SUPPRESSPKGDEP) > /dev/null

647 #
648 # The .lics files were created during pkgmogrification, and list the
649 # set of licenses to pull from $SRC for each package.  Because
650 # licenses may be duplicated between packages, we uniquify them as
651 # well as aggregating them here.
652 #
653 license-list: makesilent

```

```

654     $(PKGDEBUG){ for l in `cat $(PROC_PKGS:%.mog=%.lics)`; \
655         do print $$l; done } | sort -u > $@

657 #
658 # Staging the license and description files in the proto area allows
659 # us to do proper unreferenced file checking of both license and
660 # description files without blanket exceptions, and to pull license
661 # content without reference to $CODEMGR_WS during publication.
662 #
663 stage-licenses: license-list FRC
664     $(PKGDEBUG)$ (MAKE) -e -f Makefile.lic \
665     PKGDEBUG=$(PKGDEBUG) LICROOT=$(PKGROOT)/licenses \
666     '$(AWK)' '{ \
666     '$(NAWK)' }{ \
667         print "$(PKGROOT)/licenses/" $$0; \
668         print "$(PKGROOT)/licenses/" $$0 ".descrip"; \
669     }' license-list' > /dev/null;

671 protocmp: makesilent
672     @validate_pkg -a $(PKGSMACH) -v \
673     $(EXCEPTIONS:%=-e $(CODEMGR_WS)/exception_lists/%) \
674     -m $(PDIR) -p $(PKGROOT) -p $(TOOLSROOT)

676 pmodes: makesilent
677     @validate_pkg -a $(PKGSMACH) -M -m $(PDIR) \
678     -e $(CODEMGR_WS)/exception_lists/pmodes

680 check: protocmp pmodes

682 protolist: proto_list_$(PKGSMACH)

684 proto_list_$(PKGSMACH): $(PROC_PKGS)
685     @validate_pkg -a $(PKGSMACH) -L -m $(PDIR) > $@

687 $(PROC_PKGS): $(PDIR)

689 #
690 # This is a convenience target to allow package names to function as
691 # build targets. Generally, using it is only useful when iterating on
692 # development of a manifest.
693 #
694 # When processing a manifest, use the basename (without extension) of
695 # the package. When publishing, use the basename with a ".pub"
696 # extension.
697 #
698 # Other than during manifest development, the preferred usage is to
699 # avoid these targets and override PKGS on the make command line and
700 # use the provided all and install targets.
701 #
702 $(PKGS) $(SYNTH_PKGS): $(PDIR)/$$(@:%=%.mog)

704 $(PKGS:%=%.pub) $(SYNTH_PKGS:%=%.pub): $(PDIR)/$$(@)

706 #
707 # This is a convenience target to resolve dependencies without publishing
708 # packages.
709 #
710 gendeps: $(PDIR)/gendeps

712 #
713 # These are convenience targets for cross-platform packaging. If you
714 # want to build any of "the normal" targets for a different
715 # architecture, simply use "arch/target" as your build target.
716 #
717 # Since the most common use case for this is "install," the architecture
718 # specific install targets have been further abbreviated to elide "/install."

```

```

719 #
720 i386/% sparc/%:
721     $(MAKE) -e $(@F) PKGMACH=$(@D) SUPPRESSPKGDEP=$(SUPPRESSPKGDEP)

723 i386 sparc: $$(@)/install

725 FRC:

727 # EXPORT DELETE START
728 XMOD_PKGS= \
729     BRCMbnx \
730     BRCMbnxe \
731     SUNWadpu320 \
732     SUNWibsdpiib \
733     SUNWkdc \
734     SUNWlsimega \
735     SUNWwbint \
736     SUNWwbsup

738 EXPORT_SRC: CRYPT_SRC
739     $(RM) $(XMOD_PKGS:%=manifests/%.mf)
740     $(RM) Makefile+
741     $(SED) -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
742     < Makefile > Makefile+
743     $(MV) -f Makefile+ Makefile
744     $(CHMOD) 444 Makefile
745 # EXPORT DELETE END

```

new/usr/src/pkg/manifests/system-extended-system-utilities.mf

1

```
*****
11727 Sun May 26 18:23:39 2013
new/usr/src/pkg/manifests/system-extended-system-utilities.mf
XXX Remove nawk(1)
*****
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) 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 #
26 set name=pkg.fmri value=pkg:/system/extended-system-utilities@$(PKGVERS)
27 set name=pkg.description \
28     value="additional UNIX system utilities, including awk, bc, cal, compress, d
29 set name=pkg.summary value="Extended System Utilities"
30 set name=info.classification value=org.opensolaris.category.2008:System/Core
31 set name=variant.arch value=$(ARCH)
32 dir path=usr group=sys
33 dir path=usr/bin
34 $(i386_ONLY)dir path=usr/bin/$(ARCH32)
35 dir path=usr/bin/$(ARCH64)
36 dir path=usr/lib
37 dir path=usr/lib/$(ARCH64)
38 dir path=usr/lib/adb group=sys
39 dir path=usr/lib/adb/$(ARCH64) group=sys
40 dir path=usr/lib/fs group=sys
41 dir path=usr/lib/fs/pcfs group=sys
42 dir path=usr/lib/spell
43 dir path=usr/proc
44 dir path=usr/proc/bin
45 dir path=usr/sbin
46 dir path=usr/sbin/$(ARCH64)
47 dir path=usr/share
48 dir path=usr/share/lib
49 dir path=usr/share/lib/dict
50 dir path=usr/share/man/man1
51 dir path=usr/share/man/man1m
52 $(i386_ONLY)file path=usr/bin/$(ARCH32)/pargs mode=0555
53 $(i386_ONLY)file path=usr/bin/$(ARCH32)/pcred mode=0555
54 $(i386_ONLY)file path=usr/bin/$(ARCH32)/pfiles mode=0555
55 $(i386_ONLY)file path=usr/bin/$(ARCH32)/pflags mode=0555
56 $(i386_ONLY)file path=usr/bin/$(ARCH32)/pldd mode=0555
57 $(i386_ONLY)file path=usr/bin/$(ARCH32)/plgrp mode=0555
58 $(i386_ONLY)file path=usr/bin/$(ARCH32)/pmadvise mode=0555
59 $(i386_ONLY)file path=usr/bin/$(ARCH32)/pmap mode=0555
60 $(i386_ONLY)file path=usr/bin/$(ARCH32)/ppgsz mode=0555
61 $(i386_ONLY)file path=usr/bin/$(ARCH32)/ppriv mode=0555
```

new/usr/src/pkg/manifests/system-extended-system-utilities.mf

2

```
62 $(i386_ONLY)file path=usr/bin/$(ARCH32)/preap mode=0555
63 $(i386_ONLY)file path=usr/bin/$(ARCH32)/prun mode=0555
64 $(i386_ONLY)file path=usr/bin/$(ARCH32)/psig mode=0555
65 $(i386_ONLY)file path=usr/bin/$(ARCH32)/pstack mode=0555
66 $(i386_ONLY)file path=usr/bin/$(ARCH32)/pstop mode=0555
67 $(i386_ONLY)file path=usr/bin/$(ARCH32)/ptime mode=0555
68 $(i386_ONLY)file path=usr/bin/$(ARCH32)/ptree mode=0555
69 $(i386_ONLY)file path=usr/bin/$(ARCH32)/pwait mode=0555
70 $(i386_ONLY)file path=usr/bin/$(ARCH32)/pwdx mode=0555
71 $(i386_ONLY)file path=usr/bin/$(ARCH32)/sort mode=0555
72 file path=usr/bin/$(ARCH64)/pargs mode=0555
73 file path=usr/bin/$(ARCH64)/pcred mode=0555
74 file path=usr/bin/$(ARCH64)/pfiles mode=0555
75 file path=usr/bin/$(ARCH64)/pflags mode=0555
76 file path=usr/bin/$(ARCH64)/pldd mode=0555
77 file path=usr/bin/$(ARCH64)/plgrp mode=0555
78 file path=usr/bin/$(ARCH64)/pmadvise mode=0555
79 file path=usr/bin/$(ARCH64)/pmap mode=0555
80 file path=usr/bin/$(ARCH64)/ppgsz mode=0555
81 file path=usr/bin/$(ARCH64)/ppriv mode=0555
82 file path=usr/bin/$(ARCH64)/preap mode=0555
83 file path=usr/bin/$(ARCH64)/prun mode=0555
84 file path=usr/bin/$(ARCH64)/psig mode=0555
85 file path=usr/bin/$(ARCH64)/pstack mode=0555
86 file path=usr/bin/$(ARCH64)/pstop mode=0555
87 file path=usr/bin/$(ARCH64)/ptime mode=0555
88 file path=usr/bin/$(ARCH64)/ptree mode=0555
89 file path=usr/bin/$(ARCH64)/pwait mode=0555
90 file path=usr/bin/$(ARCH64)/pwdx mode=0555
91 file path=usr/bin/$(ARCH64)/sort mode=0555
92 file path=usr/bin/asa mode=0555
93 file path=usr/bin/awk mode=0555
94 file path=usr/bin/banner mode=0555
95 file path=usr/bin/batch mode=0555
96 file path=usr/bin/bc mode=0555
97 file path=usr/bin/bdiff mode=0755
98 file path=usr/bin/bfs mode=0555
99 file path=usr/bin/cal mode=0555
100 file path=usr/bin/calendar mode=0555
101 file path=usr/bin/col mode=0555
102 file path=usr/bin/compress mode=0555
103 file path=usr/bin/csplite mode=0555
104 file path=usr/bin/dc mode=0555
105 file path=usr/bin/diff mode=0555
106 file path=usr/bin/diff3 mode=0555
107 file path=usr/bin/dircmp mode=0555
108 file path=usr/bin/dos2unix mode=0555
109 file path=usr/bin/expand mode=0555
110 file path=usr/bin/factor mode=0555
111 file path=usr/bin/kstat mode=0555
112 file path=usr/bin/last mode=0555
113 file path=usr/bin/lastcomm mode=0555
114 file path=usr/bin/lgrpinfo mode=0555
115 file path=usr/bin/look mode=0755
116 file path=usr/bin/mkfifo mode=0555
117 file path=usr/bin/nawk mode=0555
117 file path=usr/bin/newform mode=0555
118 file path=usr/bin/news mode=0555
119 file path=usr/bin/nl mode=0555
120 file path=usr/bin/pack mode=0555
121 file path=usr/bin/pginfo mode=0555
122 file path=usr/bin/pgstat mode=0555
123 file path=usr/bin/sdiff mode=0555
124 file path=usr/bin/spell mode=0555
125 file path=usr/bin/split mode=0555
126 file path=usr/bin/tcopy mode=0555
```

```

127 file path=usr/bin/unexpand mode=0555
128 file path=usr/bin/units mode=0555
129 file path=usr/bin/unix2dos mode=0555
130 file path=usr/bin/unpack mode=0555
131 file path=usr/bin/uudecode group=uucp mode=0555
132 file path=usr/bin/uencode group=uucp mode=0555
133 file path=usr/bin/yes mode=0555
134 file path=usr/lib/$(ARCH64)/madv.so.1
135 file path=usr/lib/$(ARCH64)/mpss.so.1
136 file path=usr/lib/adb/$(ARCH64)/adbsub.o group=sys
137 file path=usr/lib/adb/adbgen group=sys mode=0755
138 file path=usr/lib/adb/adbgen1 group=sys mode=0755
139 file path=usr/lib/adb/adbgen3 group=sys mode=0755
140 file path=usr/lib/adb/adbgen4 group=sys mode=0755
141 file path=usr/lib/adb/adbsub.o group=sys
142 file path=usr/lib/diff3prog mode=0555
143 file path=usr/lib/fs/pcfs/fsck mode=0555
144 file path=usr/lib/fs/pcfs/fstyp.so.1 mode=0555
145 file path=usr/lib/fs/pcfs/mkfs mode=0555
146 file path=usr/lib/fs/pcfs/mount mode=0555
147 file path=usr/lib/madv.so.1
148 file path=usr/lib/mpss.so.1
149 file path=usr/lib/spell/compress mode=0555
150 file path=usr/lib/spell/hashcheck mode=0555
151 file path=usr/lib/spell/hashmake mode=0555
152 file path=usr/lib/spell/hlista
153 file path=usr/lib/spell/hlistb
154 file path=usr/lib/spell/hstop
155 file path=usr/lib/spell/spellin mode=0555
156 file path=usr/lib/spell/spellprog mode=0555
157 file path=usr/sbin/dmesg mode=0555
158 file path=usr/sbin/projadd group=sys mode=0555
159 file path=usr/sbin/projdel group=sys mode=0555
160 file path=usr/sbin/projmod group=sys mode=0555
161 file path=usr/share/lib/dict/words
162 file path=usr/share/man/man1/asa.1
163 file path=usr/share/man/man1/awk.1
164 file path=usr/share/man/man1/banner.1
165 file path=usr/share/man/man1/bc.1
166 file path=usr/share/man/man1/bdiff.1
167 file path=usr/share/man/man1/bfs.1
168 file path=usr/share/man/man1/cal.1
169 file path=usr/share/man/man1/calendar.1
170 file path=usr/share/man/man1/col.1
171 file path=usr/share/man/man1/compress.1
172 file path=usr/share/man/man1/cssplit.1
173 file path=usr/share/man/man1/dc.1
174 file path=usr/share/man/man1/diff.1
175 file path=usr/share/man/man1/diff3.1
176 file path=usr/share/man/man1/dircmp.1
177 file path=usr/share/man/man1/dos2unix.1
178 file path=usr/share/man/man1/expand.1
179 file path=usr/share/man/man1/factor.1
180 file path=usr/share/man/man1/hashcheck.1
181 file path=usr/share/man/man1/hashmake.1
182 file path=usr/share/man/man1/last.1
183 file path=usr/share/man/man1/lastcomm.1
184 file path=usr/share/man/man1/lgrpinfo.1
185 file path=usr/share/man/man1/look.1
186 file path=usr/share/man/man1/madv.so.1.1
187 file path=usr/share/man/man1/mpss.so.1.1
188 file path=usr/share/man/man1/nawk.1
189 file path=usr/share/man/man1/newform.1
190 file path=usr/share/man/man1/news.1
191 file path=usr/share/man/man1/nl.1
192 file path=usr/share/man/man1/pack.1

```

```

193 file path=usr/share/man/man1/pargs.1
194 file path=usr/share/man/man1/pcat.1
195 file path=usr/share/man/man1/plgrp.1
196 file path=usr/share/man/man1/pmadvice.1
197 file path=usr/share/man/man1/pmap.1
198 file path=usr/share/man/man1/ppgsz.1
199 file path=usr/share/man/man1/ppriv.1
200 file path=usr/share/man/man1/preap.1
201 file path=usr/share/man/man1/ptree.1
202 file path=usr/share/man/man1/sdiff.1
203 file path=usr/share/man/man1/sort.1
204 file path=usr/share/man/man1/spell.1
205 file path=usr/share/man/man1/spellin.1
206 file path=usr/share/man/man1/split.1
207 file path=usr/share/man/man1/tcopy.1
208 file path=usr/share/man/man1/uncompress.1
209 file path=usr/share/man/man1/unexpand.1
210 file path=usr/share/man/man1/units.1
211 file path=usr/share/man/man1/unix2dos.1
212 file path=usr/share/man/man1/unpack.1
213 file path=usr/share/man/man1/yes.1
214 file path=usr/share/man/man1/zcat.1
215 file path=usr/share/man/man1m/adbgen.1m
216 file path=usr/share/man/man1m/dmesg.1m
217 file path=usr/share/man/man1m/fsck_pcfs.1m
218 file path=usr/share/man/man1m/kstat.1m
219 file path=usr/share/man/man1m/mkfifo.1m
220 file path=usr/share/man/man1m/mkfs_pcfs.1m
221 file path=usr/share/man/man1m/mount_pcfs.1m
222 file path=usr/share/man/man1m/projadd.1m
223 file path=usr/share/man/man1m/projdel.1m
224 file path=usr/share/man/man1m/projmod.1m
225 hardlink path=usr/bin/oawk target=../../usr/bin/awk
226 hardlink path=usr/bin/pargs target=../../usr/lib/isaexec
227 hardlink path=usr/bin/pcrd target=../../usr/lib/isaexec
228 hardlink path=usr/bin/pfiles target=../../usr/lib/isaexec
229 hardlink path=usr/bin/pflags target=../../usr/lib/isaexec
230 hardlink path=usr/bin/pldd target=../../usr/lib/isaexec
231 hardlink path=usr/bin/plgrp target=../../usr/lib/isaexec
232 hardlink path=usr/bin/pmadvice target=../../usr/lib/isaexec
233 hardlink path=usr/bin/pmap target=../../usr/lib/isaexec
234 hardlink path=usr/bin/ppgsz target=../../usr/lib/isaexec
235 hardlink path=usr/bin/ppriv target=../../usr/lib/isaexec
236 hardlink path=usr/bin/preap target=../../usr/lib/isaexec
237 hardlink path=usr/bin/prun target=../../usr/lib/isaexec
238 hardlink path=usr/bin/psig target=../../usr/lib/isaexec
239 hardlink path=usr/bin/pstack target=../../usr/lib/isaexec
240 hardlink path=usr/bin/pstop target=../../usr/lib/isaexec
241 hardlink path=usr/bin/ptime target=../../usr/lib/isaexec
242 hardlink path=usr/bin/ptree target=../../usr/lib/isaexec
243 hardlink path=usr/bin/pwait target=../../usr/lib/isaexec
244 hardlink path=usr/bin/pwdx target=../../usr/lib/isaexec
245 hardlink path=usr/bin/sort target=../../usr/lib/isaexec
246 hardlink path=usr/bin/uncompress target=../../usr/bin/compress
247 hardlink path=usr/bin/zcat target=../../usr/bin/compress
248 hardlink path=usr/lib/fs/pcfs/fstyp target=../../usr/sbin/fstyp
249 legacy pkg=SUNWesu \
250 desc="additional UNIX system utilities, including awk, bc, cal, compress, di
251 name="Extended System Utilities"
252 license cr_Sun license=cr_Sun
253 license lic_CDDL license=lic_CDDL
254 license usr/src/cmd/compress/THIRDPARTYLICENSE \
255 license=usr/src/cmd/compress/THIRDPARTYLICENSE
256 license usr/src/cmd/lastcomm/THIRDPARTYLICENSE \
257 license=usr/src/cmd/lastcomm/THIRDPARTYLICENSE
258 license usr/src/cmd/look/THIRDPARTYLICENSE \

```

```
259     license=usr/src/cmd/look/THIRDPARTYLICENSE
260 license usr/src/cmd/units/THIRDPARTYLICENSE \
261     license=usr/src/cmd/units/THIRDPARTYLICENSE
262 link path=usr/bin/dmesg target=../sbin/dmesg
263 link path=usr/bin/pcat target=../unpack
264 link path=usr/bin/strace target=../sbin/strace
265 link path=usr/dict target=../share/lib/dict
266 link path=usr/proc/bin/pcred target=../bin/pcred
267 link path=usr/proc/bin/pfiles target=../bin/pfiles
268 link path=usr/proc/bin/pflags target=../bin/pflags
269 link path=usr/proc/bin/pldd target=../bin/pldd
270 link path=usr/proc/bin/pmap target=../bin/pmap
271 link path=usr/proc/bin/prun target=../bin/prun
272 link path=usr/proc/bin/psig target=../bin/psig
273 link path=usr/proc/bin/pstack target=../bin/pstack
274 link path=usr/proc/bin/pstop target=../bin/pstop
275 link path=usr/proc/bin/ptime target=../bin/ptime
276 link path=usr/proc/bin/ptree target=../bin/ptree
277 link path=usr/proc/bin/pwait target=../bin/pwait
278 link path=usr/proc/bin/pwdx target=../bin/pwdx
279 depend fmri=runtime/perl-510 type=require
```


new/usr/src/test/zfs-tests/include/commands.cfg

1

```
*****
5077 Sun May 26 18:23:39 2013
new/usr/src/test/zfs-tests/include/commands.cfg
XXX Remove awk(1)
*****
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 at
9 # http://www.illumos.org/license/CDDL.
10 #
11 #
12 #
13 # Copyright (c) 2012 by Delphix. All rights reserved.
14 #
15 #
16 export ARP="/usr/sbin/arp"
17 export AWK="/usr/bin/awk"
18 export BASENAME="/usr/bin/basename"
19 export BC="/usr/bin/bc"
20 export BUNZIP2="/usr/bin/bunzip2"
21 export BZCAT="/usr/bin/bzcat"
22 # bzip2, bunzip2 and bzcat prepend the contents of $BZIP2 or $BZIP to any
23 # arguments. Setting BZIP2 here will lead to hard-to-diagnose unhappiness.
24 export CAT="/usr/bin/cat"
25 export CD="/usr/bin/cd"
26 export CHGRP="/usr/bin/chgrp"
27 export CHMOD="/usr/bin/chmod"
28 export CHOWN="/usr/bin/chown"
29 export CKSUM="/usr/bin/cksum"
30 export CLRI="/usr/sbin/clri"
31 export CMP="/usr/bin/cmp"
32 export COMPRESS="/usr/bin/compress"
33 export COREADM="/usr/bin/coreadm"
34 export CP="/usr/bin/cp"
35 export CPIO="/usr/bin/cpio"
36 export CUT="/usr/bin/cut"
37 export DATE="/usr/bin/date"
38 export DD="/usr/bin/dd"
39 export DEVFSADM="/usr/sbin/devfsadm"
40 export DF="/usr/bin/df"
41 export DIFF="/usr/bin/diff"
42 export DIRCMP="/usr/bin/dircmp"
43 export DIRNAME="/usr/bin/dirname"
44 export DU="/usr/bin/du"
45 export DUMPADM="/usr/sbin/dumpadm"
46 export ECHO="/usr/bin/echo"
47 export EGREP="/usr/bin/egrep"
48 # Don't use $ENV here, because in ksh scripts it evaluates to
49 # $HOME/.kshrc - likely not what you wanted.
50 export FDISK="/usr/sbin/fdisk"
51 export FF="/usr/sbin/ff"
52 export FGREP="/usr/bin/fgrep"
53 export FILE="/usr/bin/file"
54 export FIND="/usr/bin/find"
55 export FMADM="/usr/sbin/fmadm"
56 export FMDUMP="/usr/sbin/fmdump"
57 export FORMAT="/usr/sbin/format"
58 export FSCK="/usr/sbin/fsck"
59 export FSDB="/usr/sbin/fsdb"
60 export FSIRAND="/usr/sbin/fsirand"
61 export FSTYP="/usr/sbin/fstyp"
```

new/usr/src/test/zfs-tests/include/commands.cfg

2

```
62 export GETENT="/usr/bin/getent"
63 export GETMAJOR="/usr/sbin/getmajor"
64 export GNUDD="/usr/gnu/bin/dd"
65 export GREP="/usr/bin/grep"
66 export GROUPADD="/usr/sbin/groupadd"
67 export GROUPDEL="/usr/sbin/groupdel"
68 export GROUPMOD="/usr/sbin/groupmod"
69 export GROUPS="/usr/bin/groups"
70 export HEAD="/usr/bin/head"
71 export HOSTNAME="/usr/bin/hostname"
72 export ID="/usr/bin/id"
73 export ISAINFO="/usr/bin/isainfo"
74 export ISCSIADM="/usr/sbin/iscsiadm"
75 export KILL="/usr/bin/kill"
76 export KSH="/usr/bin/ksh"
77 export KSTAT="/usr/bin/kstat"
78 export LABELIT="/usr/sbin/labelit"
79 export LOCKFS="/usr/sbin/lockfs"
80 export LOFIADM="/usr/sbin/lofiadm"
81 export LOGNAME="/usr/bin/logname"
82 export LS="/usr/bin/ls"
83 export MD5SUM="/usr/bin/md5sum"
84 export MDB="/usr/bin/mdb"
85 export METACLEAR="/usr/sbin/metaclear"
86 export METADB="/usr/sbin/metadb"
87 export METAINIT="/usr/sbin/metainit"
88 export METASTAT="/usr/sbin/metastat"
89 export MKDIR="/usr/bin/mkdir"
90 export MKFILE="/usr/sbin/mkfile"
91 export MKNOD="/usr/sbin/mknod"
92 export MODINFO="/usr/sbin/modinfo"
93 export MODUNLOAD="/usr/sbin/modunload"
94 export MOUNT="/usr/sbin/mount"
95 export MV="/usr/bin/mv"
96 export AWK="/usr/xpg4/bin/awk"
96 export NAWK="/usr/bin/nawk"
97 export NCHECK="/usr/sbin/ncheck"
98 export NEWFS="/usr/sbin/newfs"
99 export PACK="/usr/bin/pack"
100 export PAGESIZE="/usr/bin/pagesize"
101 export PAX="/usr/bin/pax"
102 export PFEXEC="/usr/bin/pfexec"
103 export PGREP="/usr/bin/pgrep"
104 export PING="/usr/sbin/ping"
105 export PKGINFO="/usr/bin/pkginfo"
106 export PKILL="/usr/bin/pkill"
107 export PPRIV="/usr/bin/ppriv"
108 export PRINTF="/usr/bin/printf"
109 export PRTVTOC="/usr/sbin/prtvtoctoc"
110 export PS="/usr/bin/ps"
111 export PSRINFO="/usr/sbin/psrinfo"
112 export PWD="/usr/bin/pwd"
113 export PYTHON="/usr/bin/python"
114 export QUOTAON="/usr/sbin/quotaon"
115 export RCP="/usr/bin/rcp"
116 export REBOOT="/usr/sbin/reboot"
117 export RM="/usr/bin/rm"
118 export RMDIR="/usr/bin/rmdir"
119 export RSH="/usr/bin/rsh"
120 export RUNAT="/usr/bin/runat"
121 export SED="/usr/bin/sed"
122 export SHARE="/usr/sbin/share"
123 export SHUF="/usr/bin/shuf"
124 export SLEEP="/usr/bin/sleep"
125 export SORT="/usr/bin/sort"
126 export STRINGS="/usr/bin/strings"
```

```
127 export SU="/usr/bin/su"
128 export SUM="/usr/bin/sum"
129 export SVCADM="/usr/sbin/svcadm"
130 export SVCS="/usr/bin/svcs"
131 export SWAP="/usr/sbin/swap"
132 export SWAPADD="/sbin/swapadd"
133 export SYNC="/usr/bin/sync"
134 export TAIL="/usr/bin/tail"
135 export TAR="/usr/sbin/tar"
136 export TOUCH="/usr/bin/touch"
137 export TR="/usr/bin/tr"
138 export TRUE="/usr/bin/true"
139 export TUNEFSS="/usr/sbin/tunefs"
140 export UFSDUMP="/usr/sbin/ufsdump"
141 export UFSRESTORE="/usr/sbin/ufsrestore"
142 export UMASK="/usr/bin/umask"
143 export UMOUNT="/usr/sbin/umount"
144 export UNAME="/usr/bin/uname"
145 export UNCOMPRESS="/usr/bin/uncompress"
146 export UNIQU="/usr/bin/uniq"
147 export UNPACK="/usr/bin/unpack"
148 export UNSHARE="/usr/sbin/unshare"
149 export USERADD="/usr/sbin/useradd"
150 export USERDEL="/usr/sbin/userdel"
151 export USERMOD="/usr/sbin/usermod"
152 export WAIT="/usr/bin/wait"
153 export WC="/usr/bin/wc"
154 export ZDB="/usr/sbin/zdb"
155 export ZFS="/usr/sbin/zfs"
156 export ZHACK="/usr/sbin/zhack"
157 export ZLOGIN="/usr/sbin/zlogin"
158 export ZONEADM="/usr/sbin/zoneadm"
159 export ZONECFG="/usr/sbin/zonecfg"
160 export ZONENAME="/usr/bin/zonename"
161 export ZPOOL="/usr/sbin/zpool"
```

```

new/usr/src/test/zfs-tests/include/libtest.shlib 1
*****
54467 Sun May 26 18:23:40 2013
new/usr/src/test/zfs-tests/include/libtest.shlib
XXX Remove nawk(1)
*****
_____unchanged_portion_omitted_____

665 #
666 # Get the end cyl of the given slice
667 #
668 function get_endslice #<disk> <slice>
669 {
670     typeset disk=$1
671     typeset slice=$2
672     if [[ -z $disk || -z $slice ]] ; then
673         log_fail "The disk name or slice number is unspecified."
674     fi

676     disk=${disk#/dev/dsk/}
677     disk=${disk#/dev/rdisk/}
678     disk=${disk%s*}

680     typeset -i ratio=0
681     ratio=$(($PRTVTOC /dev/rdisk/${disk}s2 | \
682         $GREP "sectors\cylinder" | \
683         $AWK '{print $2}')

685     if ((ratio == 0)); then
686         return
687     fi

689     typeset -i endcyl=$(($PRTVTOC -h /dev/rdisk/${disk}s2 |
690         $AWK -v token="$slice" '{if ($1==token) print $6}')
690     $NAWK -v token="$slice" '{if ($1==token) print $6}'

692     ((endcyl = (endcyl + 1) / ratio))
693     echo $endcyl
694 }
_____unchanged_portion_omitted_____

1343 #
1344 # Given a pool, and this function list all disks in the pool
1345 #
1346 function get_disklist # pool
1347 {
1348     typeset disklist=""

1350     disklist=$(($ZPOOL iostat -v $1 | $AWK '(NR >4) {print $1}' | \
1350     disklist=$(($ZPOOL iostat -v $1 | $NAWK '(NR >4) {print $1}' | \
1351     $GREP -v "\-\\-\-\\-\-" | \
1352     $EGREP -v -e "^(mirror|raidz1|raidz2|spare|log|cache)$")

1354     $ECHO $disklist
1355 }
_____unchanged_portion_omitted_____

1460 #
1461 # Check the output of 'zpool status -v <pool>',
1462 # and to see if the content of <token> contain the <keyword> specified.
1463 #
1464 # Return 0 is contain, 1 otherwise
1465 #
1466 function check_pool_status # pool token keyword
1467 {
1468     typeset pool=$1
1469     typeset token=$2

```

```

new/usr/src/test/zfs-tests/include/libtest.shlib 2

1470     typeset keyword=$3

1472     $ZPOOL status -v "$pool" 2>/dev/null | $AWK -v token="$token:" '
1472     $ZPOOL status -v "$pool" 2>/dev/null | $NAWK -v token="$token:" '
1473     ($1==token) {print $0}' \
1474     | $GREP -i "$keyword" > /dev/null 2>&1

1476     return $?
1477 }
_____unchanged_portion_omitted_____

1665 #/**
1666 # A function to find and locate free disks on a system or from given
1667 # disks as the parameter. It works by locating disks that are in use
1668 # as swap devices, SVM devices, and dump devices, and also disks
1669 # listed in /etc/vfstab
1670 #
1671 # @$ given disks to find which are free, default is all disks in
1672 # the test system
1673 #
1674 # @return a string containing the list of available disks
1675 #*/
1676 function find_disks
1677 {
1678     sfi=/tmp/swaplist.$$
1679     msi=/tmp/metastat.$$
1680     dmpi=/tmp/dumpdev.$$
1681     max_findedisksnum=${MAX_FINDDISKSNUM:-6}

1683     $SWAP -l > $sfi
1684     $METASTAT -c > $msi 2>/dev/null
1685     $DUMPADM > $dmpi 2>/dev/null

1687 # write an awk script that can process the output of format
1688 # to produce a list of disks we know about. Note that we have
1689 # to escape "$2" so that the shell doesn't interpret it while
1690 # we're creating the awk script.
1691 # -----
1692     $CAT > /tmp/find_disks.awk <<EOF
1693     #!/usr/xpg4/bin/awk -f
1693     #!/bin/nawk -f
1694     BEGIN { FS="."; }

1696     /^Specify disk/{
1697         searchdisks=0;
1698     }

1700     {
1701         if (searchdisks && $2 !~ "^$"){
1702             split($2,arr," ");
1703             print arr[1];
1704         }
1705     }

1707     /^AVAILABLE DISK SELECTIONS:/{
1708         searchdisks=1;
1709     }
1710 EOF
1711 #-----

1713     $CHMOD 755 /tmp/find_disks.awk
1714     disks=${@:-$(($ECHO " " | $FORMAT -e 2>/dev/null | /tmp/find_disks.awk)}
1715     $RM /tmp/find_disks.awk

1717     unused=""
1718     for disk in $disks; do

```

```

1719 # Check for mounted
1720 $GREP "${disk}[sp]" /etc/mnttab >/dev/null
1721 (( $? == 0 )) && continue
1722 # Check for swap
1723 $GREP "${disk}[sp]" $sfi >/dev/null
1724 (( $? == 0 )) && continue
1725 # Check for SVM
1726 $GREP "${disk}" $msi >/dev/null
1727 (( $? == 0 )) && continue
1728 # check for dump device
1729 $GREP "${disk}[sp]" $dmpi >/dev/null
1730 (( $? == 0 )) && continue
1731 # check to see if this disk hasn't been explicitly excluded
1732 # by a user-set environment variable
1733 $ECHO "${ZFS_HOST_DEVICES_IGNORE}" | $GREP "${disk}" > /dev/null
1734 (( $? == 0 )) && continue
1735 unused_candidates="$unused_candidates $disk"
1736 done
1737 $RM $sfi
1738 $RM $msi
1739 $RM $dmpi

```

```

1741 # now just check to see if those disks do actually exist
1742 # by looking for a device pointing to the first slice in
1743 # each case. limit the number to max_finddisksnum
1744 count=0
1745 for disk in $unused_candidates; do
1746     if [ -b /dev/dsk/${disk}s0 ]; then
1747         if [ $count -lt $max_finddisksnum ]; then
1748             unused="$unused $disk"
1749             # do not impose limit if $@ is provided
1750             [[ -z $@ ]] && ((count = count + 1))
1751         fi
1752     fi
1753 done

```

```

1755 # finally, return our disk list
1756 $ECHO $unused
1757 }

```

unchanged_portion_omitted

```

2320 #
2321 # Get the given disk/slice state from the specific field of the pool
2322 #
2323 function get_device_state #pool disk field("","spares","logs")
2324 {
2325     typeset pool=$1
2326     typeset disk=${2#/dev/dsk/}
2327     typeset field=${3:-$pool}
2328
2329     state=$(ZPOOL status -v "$pool" 2>/dev/null | \
2330 $AWK -v device=$disk -v pool=$pool -v field=$field \
2331 $NAWK -v device=$disk -v pool=$pool -v field=$field \
2332 'BEGIN {startconfig=0; startfield=0;}
2333 /config:/ {startconfig=1}
2334 (startconfig==1) && ($1==field) {startfield=1; next;}
2335 (startfield==1) && ($1==device) {print $2; exit;}
2336 (startfield==1) &&
2337 ($1==field || $1 ~ "^spares$" || $1 ~ "^logs$") {startfield=0}')
2338     echo $state
2339 }

```

unchanged_portion_omitted

```

2462 #
2463 # Get the sub string from specified source string
2464 #

```

```

2465 # $1 source string
2466 # $2 start position. Count from 1
2467 # $3 offset
2468 #
2469 function get_substr #src_str pos offset
2470 {
2471     typeset pos offset

```

```

2473     $ECHO $1 | \
2474 $AWK -v pos=$2 -v offset=$3 '{print substr($0, pos, offset)}'
2475 $NAWK -v pos=$2 -v offset=$3 '{print substr($0, pos, offset)}'
2476 }

```

unchanged_portion_omitted

new/usr/src/test/zfs-tests/tests/functional/acl/acl_common.kshlib 1

12186 Sun May 26 18:23:42 2013

new/usr/src/test/zfs-tests/tests/functional/acl/acl_common.kshlib

XXX Remove awk(1)

unchanged portion omitted

```
45 #
46 # Get the given file/directory ACL
47 #
48 # $1 object -- file or directroy
49 #
50 function get_acl #<obj>
51 {
52     typeset obj=$1
53     if (( ${#obj} == 0 )); then
54         return 1
55     fi
```

```
57     $LS -vd $obj | $AWK '(NR != 1) {print $0}'
57     $LS -vd $obj | $NAWK '(NR != 1) {print $0}'
58 }
```

```
60 #
61 # Get the given file/directory ACL
62 #
63 # $1 object -- file or directroy
64 #
65 function get_compact_acl #<obj>
66 {
67     typeset obj=$1
68     if (( ${#obj} == 0 )); then
69         return 1
70     fi
```

```
72     $LS -vd $obj | $AWK '(NR != 1) {print $0}'
72     $LS -vd $obj | $NAWK '(NR != 1) {print $0}'
73 }
```

unchanged portion omitted

```
189 #
190 # Check '+' is set for a given file/directory with 'ls [-v]' command
191 #
192 # $1 object -- file or directory.
193 #
194 function plus_sign_check_v #<obj>
195 {
196     typeset obj=$1
197     if (( ${#obj} == 0 )); then
198         return 1
199     fi
```

```
201     $LS -vd $obj | $AWK '(NR == 1) {print $1}' | $GREP "+\>" > /dev/null
201     $LS -vd $obj | $NAWK '(NR == 1) {print $1}' | $GREP "+\>" > /dev/null
```

```
203     return $?
204 }
```

unchanged portion omitted

```
239 #
240 # Count how many ACEs for the specied file or directory.
241 #
242 # $1 file or directroy name
243 #
244 function count_ACE #<file or dir name>
245 {
```

new/usr/src/test/zfs-tests/tests/functional/acl/acl_common.kshlib 2

```
246     if [[ ! -e $1 ]]; then
247         log_note "Need input file or directroy name."
248         return 1
249     fi
```

```
251     $LS -vd $1 | $AWK 'BEGIN {count=0}
251     $LS -vd $1 | $NAWK 'BEGIN {count=0}
252         (NR != 1)&&(/[0-9]:/) {count++}
253         END {print count}'
```

```
255     return 0
256 }
```

unchanged portion omitted

```
new/usr/src/test/zfs-tests/tests/functional/cli_root/zfs_send/zfs_send_006_pos.k 1
```

```
*****
```

```
5917 Sun May 26 18:23:44 2013
```

```
new/usr/src/test/zfs-tests/tests/functional/cli_root/zfs_send/zfs_send_006_pos.k
```

```
sh
```

```
XXX Remove nawk(1)
```

```
*****
```

```
unchanged_portion_omitted
```

```
74 function verify_size_estimates
75 {
76     typeset options=$1
77     typeset file_size=$2
78     typeset refer_diff=$((ECHO "$refer_size - $estimate_size" | bc)
79     refer_diff=$((ECHO "$refer_diff / 1" | bc)
80     refer_diff=$((ECHO "$refer_diff" | $AWK '{print ($1 < 0) ? ($1 * -1): $1}
80     refer_diff=$((ECHO "$refer_diff" | $NAWK '{print ($1 < 0) ? ($1 * -1): $
81     typeset file_diff=$((ECHO "$file_size - $estimate_size" | bc)
82     file_diff=$((ECHO "$file_diff / 1" | bc)
83     file_diff=$((ECHO "$file_diff" | $AWK '{print ($1 < 0) ? ($1 * -1):$1'})
83     file_diff=$((ECHO "$file_diff" | $NAWK '{print ($1 < 0) ? ($1 * -1):$1'})
84     typeset expected_diff=$((cal_percentage $refer_size)

86     [[ -z $refer_diff && -z $file_diff && -z $expected_diff ]] && \
87     log_fail "zfs send $options failed"
88     [[ $refer_diff -le $expected_diff && \
89     $file_diff -le $expected_diff ]] || \
90     log_fail "zfs send $options gives wrong size estimates"
91 }
```

```
unchanged_portion_omitted
```

new/usr/src/test/zfs-tests/tests/functional/cli_root/zfs_snapshot/zfs_snapshot_0 1

```
*****
3399 Sun May 26 18:23:45 2013
new/usr/src/test/zfs-tests/tests/functional/cli_root/zfs_snapshot/zfs_snapshot_0
09_pos.ksh
XXX Remove awk(1)
*****
unchanged portion omitted
36 datasets="$TESTPOOL/$TESTFS1 $TESTPOOL/$TESTFS2
37 $TESTPOOL/$TESTFS3"

39 invalid_args=(" $TESTPOOL/$TESTFS1@now $TESTPOOL/$TESTFS2@now \
40 $TESTPOOL/$TESTFS@blah? " $TESTPOOL/$TESTFS1@blah* \
41 $TESTPOOL/$TESTFS2@blah? $TESTPOOL/$TESTFS3@blah% " \
42 " $TESTPOOL/$TESTFS1@$( $PYTHON -c 'print "x" * 300' ) $TESTPOOL/$TESTFS2@300 \
43 $TESTPOOL/$TESTFS3@300" )

45 valid_args=(" $TESTPOOL/$TESTFS1@snap $TESTPOOL/$TESTFS2@snap \
46 $TESTPOOL/$TESTFS3@snap " $TESTPOOL/$TESTFS1@$( $PYTHON -c 'print "x" * 200' )
47 $TESTPOOL/$TESTFS2@2 $TESTPOOL/$TESTFS3@s" )

49 log_assert "verify zfs supports multiple consistent snapshots"
50 log_onexit cleanup
51 typeset -i i=1
52 test_data=$STF_SUITE/tests/functional/cli_root/zpool_upgrade/blockfiles

54 log_note "destroy a list of valid snapshots"
55 for ds in $datasets; do
56     log_must $ZFS create $ds
57     log_must $CP -r $test_data /$ds
58 done
59 i=0
60 while (( i < ${#valid_args[*]} )); do
61     log_must $ZFS snapshot ${valid_args[i]}
62     for token in ${valid_args[i]}; do
63         log_must snapexists $token && \
64             log_must $ZFS destroy $token
65     done
66     ((i = i + 1))
67 done
68 log_note "destroy a list of invalid snapshots"
69 i=0
70 while (( i < ${#invalid_args[*]} )); do
71     log_mustnot $ZFS snapshot ${invalid_args[i]}
72     for token in ${invalid_args[i]}; do
73         log_mustnot snapexists $token
74     done
75     ((i = i + 1))
76 done
77 log_note "verify multiple snapshot transaction group"
78 txg_group=$( $ZDB -Pd $TESTPOOL | $GREP snap | $AWK '{print $7}' )
79 for i in 1 2 3; do
80     txg_tag=$( $ECHO "$txg_group" | $AWK -v j=$i 'FNR == j {print}' )
80     txg_tag=$( $ECHO "$txg_group" | $NAWK -v j=$i 'FNR == j {print}' )
81     [[ $txg_tag != $( $ECHO "$txg_group" | \
82         $AWK -v j=$i 'FNR == j {print}' ) ]] \
82     $NAWK -v j=$i 'FNR == j {print}' ) ]] \
83     && log_fail "snapshots belong to differnt transaction groups"
84 done
85 log_note "verify snapshot contents"
86 for ds in $datasets; do
87     status=$( $DIRCMP /$ds /$ds/.zfs/snapshot/snap | $GREP "different" )
88     [[ -z $status ]] || log_fail "snapshot contents are different from" \
89     "the filesystem"
90 done

92 log_note "verify multiple snapshot with -r option"
```

new/usr/src/test/zfs-tests/tests/functional/cli_root/zfs_snapshot/zfs_snapshot_0 2

```
93 log_must $ZFS create $TESTPOOL/TESTFS4
94 log_must $ZFS create -p $TESTPOOL/$TESTFS3/TESTFSA$( $PYTHON -c 'print "x" * 210' )
95 log_mustnot $ZFS snapshot -r $TESTPOOL/$TESTFS1@snap1 $TESTPOOL/$TESTFS2@snap1 \
96     $TESTPOOL/$TESTFS3@snap1 $TESTPOOL/TESTFS4@snap1
97 log_must $ZFS rename $TESTPOOL/$TESTFS3/TESTFSA$( $PYTHON -c 'print "x" * 210' )
98     $TESTPOOL/$TESTFS3/TESTFSA
99 log_must $ZFS snapshot -r $TESTPOOL/$TESTFS1@snap1 $TESTPOOL/$TESTFS2@snap1 \
100     $TESTPOOL/$TESTFS3@snap1 $TESTPOOL/TESTFS4@snap1

102 log_pass "zfs multiple snapshot verified correctly"
```

new/usr/src/test/zfs-tests/tests/functional/cli_root/zfs_upgrade/zfs_upgrade_001 1

```
*****
4211 Sun May 26 18:23:45 2013
new/usr/src/test/zfs-tests/tests/functional/cli_root/zfs_upgrade/zfs_upgrade_001
_pos.ksh
XXX Remove awk(1)
*****
_____unchanged_portion_omitted_____

62 log_assert "Executing 'zfs upgrade' command succeeds."
63 log_onexit cleanup

65 rootfs=$TESTPOOL/$TESTFS
66 typeset output=/tmp/zfs-versions.$$
67 typeset oldoutput=/tmp/zfs-versions-old.$$
68 typeset expect_str1="This system is currently running ZFS filesystem version"
69 typeset expect_str2="All filesystems are formatted with the current version"
70 typeset expect_str3="The following filesystems are out of date, and can be upgra
71 typeset -i COUNT OLDCOUNT

73 $ZFS upgrade | $AWK '$1 ~ "[0-9]+$" {print $2}'> $oldoutput
73 $ZFS upgrade | $NAWK '$1 ~ "[0-9]+$" {print $2}'> $oldoutput
74 OLDCOUNT=$( $WC -l $oldoutput | $AWK '{print $1}' )

76 old_datasets=""
77 for version in $ZFS_ALL_VERSIONS ; do
78     typeset verfs
79     eval verfs=\$ZFS_VERSION_$version
80     typeset current_fs=$rootfs/$verfs
81     typeset current_snap=${current_fs}@snap
82     typeset current_clone=$rootfs/clone$verfs
83     log_must $ZFS create -o version=${version} ${current_fs}
84     log_must $ZFS snapshot ${current_snap}
85     log_must $ZFS clone ${current_snap} ${current_clone}

87     if (( version != $ZFS_VERSION )) ; then
88         old_datasets="$old_datasets ${current_fs} ${current_clone}"
89     fi
90 done

92 if is_global_zone ; then
93     log_must $ZFS create -V 100m $rootfs/$TESTVOL
94 fi

96 log_must eval '$ZFS upgrade > $output 2>&1'

98 # we also check that the usage message contains at least a description
99 # of the current ZFS version.
100 log_must eval '$GREP "${expect_str1} $ZFS_VERSION" $output > /dev/null 2>&1'
101 $ZFS upgrade | $AWK '$1 ~ "[0-9]+$" {print $2}'> $output
101 $ZFS upgrade | $NAWK '$1 ~ "[0-9]+$" {print $2}'> $output
102 COUNT=$( $WC -l $output | $AWK '{print $1}' )

104 typeset -i i=0
105 for fs in ${old_datasets} ; do
106     log_must $GREP "^$fs$" $output
107     (( i = i + 1 ))
108 done

110 if (( i != COUNT - OLDCOUNT )) ; then
111     $CAT $output
112     log_fail "More old-version filesystems print out than expect."
113 fi

115 for fs in $old_datasets ; do
116     if datasetexists $fs ; then
117         log_must $ZFS destroy -Rf $fs
```

new/usr/src/test/zfs-tests/tests/functional/cli_root/zfs_upgrade/zfs_upgrade_001 2

```
118     fi
119 done

121 log_must eval '$ZFS upgrade > $output 2>&1'
122 log_must eval '$GREP "${expect_str1} $ZFS_VERSION" $output > /dev/null 2>&1'
123 if (( OLDCOUNT == 0 )) ; then
124     log_must eval '$GREP "${expect_str2}" $output > /dev/null 2>&1'
125 else
126     log_must eval '$GREP "${expect_str3}" $output > /dev/null 2>&1'
127 fi
128 $ZFS upgrade | $AWK '$1 ~ "[0-9]+$" {print $2}'> $output
128 $ZFS upgrade | $NAWK '$1 ~ "[0-9]+$" {print $2}'> $output
129 COUNT=$( $WC -l $output | $AWK '{print $1}' )

131 if (( COUNT != OLDCOUNT )) ; then
132     $CAT $output
133     log_fail "Unexpected old-version filesystems print out."
134 fi

136 log_pass "Executing 'zfs upgrade' command succeeds."
```


new/usr/src/test/zfs-tests/tests/functional/cli_root/zfs_upgrade/zfs_upgrade_002 1

1892 Sun May 26 18:23:46 2013

new/usr/src/test/zfs-tests/tests/functional/cli_root/zfs_upgrade/zfs_upgrade_002_pos.ksh

XXX Remove `awk(1)`

unchanged_portion_omitted

50 log_assert "Executing 'zfs upgrade -v' command succeeds."
51 log_onexit cleanup

53 typeset output=/tmp/zfs-versions.\$\$
54 typeset expect_str1="Initial ZFS filesystem version"
55 typeset expect_str2="Enhanced directory entries"

57 log_must eval '\$ZFS upgrade -v > /dev/null 2>&1'

59 `$ZFS upgrade -v | $AWK '$1 ~ "[0-9]+$" {print $0}'> $output`
59 `$ZFS upgrade -v | $NAWK '$1 ~ "[0-9]+$" {print $0}'> $output`
60 log_must eval '\$GREP "\${expect_str1}" \$output > /dev/null 2>&1'
61 log_must eval '\$GREP "\${expect_str2}" \$output > /dev/null 2>&1'

63 log_pass "Executing 'zfs upgrade -v' command succeeds."

new/usr/src/tools/ndrgen/ndrgen.sh

1

2946 Sun May 26 18:23:46 2013

new/usr/src/tools/ndrgen/ndrgen.sh

XXX Remove awk(1)

unchanged_portion_omitted_

```
45 # Copy header text from the input ndl file to the generated ndr C file.
46 ndrgen_copy_header()
47 {
48     ndl_file=$1
49     ndr_file=$2
```

```
51     /usr/xpg4/bin/awk 'BEGIN { copy=0; }
51     awk 'BEGIN { copy=0; }
52     /^\\* NDRGEN_HEADER_BEGIN \\*\\/ { copy=1; next; }
53     /^\\* NDRGEN_HEADER_END \\*\\/ { copy=0; next; }
54     ./ { if (copy==1) print; }' < $ndl_file > $ndr_file
55 }
```

unchanged_portion_omitted_

new/usr/src/tools/scripts/Install.sh

1

```
*****
25651 Sun May 26 18:23:46 2013
new/usr/src/tools/scripts/Install.sh
XXX Remove awk(1)
*****
_____unchanged_portion_omitted_____

232 #
233 # Patch up target directories for glommed kernel.
234 # usage: fixglom listfile glomname
235 #

237 function fixglom {
238     /usr/xpg4/bin/awk \
238     nawk \
239     -v glomname=$2 \
240     -v karch=$KARCH '
241     $1 == "MOD" || $1 == "SYMLINK" {
242         sub(/^platform.*kernel/, "platform/" karch "/" glomname, $4)
243         sub(/^kernel/, "platform/" karch "/" glomname, $4)
244         sub(/^usr.kernel/, "platform/" karch "/" glomname, $4)
245         print
246     }
247     $1 == "LINK" {
248         sub(/^platform.*kernel/, "platform/" karch "/" glomname, $3)
249         sub(/^kernel/, "platform/" karch "/" glomname, $3)
250         sub(/^usr.kernel/, "platform/" karch "/" glomname, $3)
251         sub(/^platform.*kernel/, "platform/" karch "/" glomname, $5)
252         sub(/^kernel/, "platform/" karch "/" glomname, $5)
253         sub(/^usr.kernel/, "platform/" karch "/" glomname, $5)
254         print
255     }
256     $1 == "CONF" {
257         sub(/^platform.*kernel/, "platform/" karch "/" glomname, $3)
258         sub(/^kernel/, "platform/" karch "/" glomname, $3)
259         sub(/^usr.kernel/, "platform/" karch "/" glomname, $3)
260         print
261     }
262     ' $1 > $1.new
263     mv $1.new $1
264 }

266 #
267 # Filter out implementation-specific modules, unless that
268 # implementation was requested by the user.
269 # usage: filtimpl listfile implname
270 #

272 function filtimpl {
273     /usr/xpg4/bin/awk \
273     nawk \
274     -v impl=$2 '
275     $1 == "MOD" || $1 == "SYMLINK" {
276         if ($6 == "all" || $6 == impl)
277             print
278     }
279     $1 == "CONF" {
280         if ($5 == "all" || $5 == impl)
281             print
282     }
283     $1 == "LINK" {
284         if ($7 == "all" || $7 == impl)
285             print
286     }
287     ' $1 > $1.new
288     mv $1.new $1
```

new/usr/src/tools/scripts/Install.sh

2

```
289 }

291 #
292 # Filter the module list to match the user's request.
293 # Usage: filtmod listfile modules
294 #
295 function filtmod {
296     /usr/xpg4/bin/awk -v reqstring="$2" '
296     nawk -v reqstring="$2" '
297     function modmatch(modname) {
298         if (reqstring == "All") {
299             return (1)
300         } else if (reqstring == "Modules") {
301             if (modname != "unix" && modname != "genunix")
302                 return (1)
303         } else {
304             if (modname in reqmods)
305                 return (1)
306         }
307         return (0)
308     }
309     BEGIN {
310         #
311         # The split call creates indexes 1, 2, 3, ... We want
312         # the module names as indexes.
313         #
314         split(reqstring, tmpmods)
315         for (i in tmpmods)
316             reqmods[tmpmods[i]] = 1
317     }
318     $1 == "MOD" {
319         if (modmatch($3))
320             print
321     }
322     $1 == "CONF" {
323         if (modmatch($6))
324             print
325     }
326     $1 == "SYMLINK" {
327         if (modmatch($7))
328             print
329     }
330     $1 == "LINK" {
331         if (modmatch($4))
332             print
333     }
334     ' $1 > $1.new
335     mv $1.new $1
336 }
_____unchanged_portion_omitted_____

454 # Sanity-check the given module list.
455 function check_modlist {
456     /usr/xpg4/bin/awk '
456     nawk '
457     BEGIN {
458         nfields["MOD"] = 6
459         nfields["CONF"] = 6
460         nfields["LINK"] = 7
461         nfields["SYMLINK"] = 7
462     }
463     {
464         # This also catches unknown tags.
465         if (nfields[$1] != NF) {
466             print "error: invalid modlist record:"
467             print $0
```

new/usr/src/tools/scripts/Install.sh

```

468             print "expected", nfields[$1], "fields, found", NF
469             status=1
470         }
471     }
472     END {
473         exit status
474     }
475     ' $1 || fail "Errors in kernel module list"
476 }

```

unchanged portion omitted

```

880 #
881 # Process options
882 #

```

```

884 RCOPTS=""
885 LIBCREATE="no"
886 LIBSRC=""
887 ENV_PATH=$CODEMGR_WS
888 OBJD="debug"
889 KMDB="yes"

```

```

891 test -s $INSTALL_RC && RCOPTS='cat $INSTALL_RC'
892 set $INSTALL $DEFAULT_OPTIONS $RCOPTS $*
893 shift

```

```

895 while getopts acd:D:G:hi:k:Kl:Lmno:pPqRs:t:T:uvVw:xX36 opt
896 do
897     case $opt in
898         w) ENV_PATH="$OPTARG"; SRC="$ENV_PATH/usr/src";;
899         s) UTS="$OPTARG";;
900         k) KARCH="$OPTARG";;
901         t|T) TARGET="$OPTARG"; IMODE=$opt; CLEANUP="c";;
902         n) TARGET=""; IMODE="n"; CLEANUP="p";;
903         u) files="unix genunix";;
904         m) files="Modules";;
905         a) files="All";;
906         v|V|q) VERBOSE=$opt;;
907         c|p) CLEANUP=$opt;;
908         L) LIBCREATE="yes"; CLEANUP="c";;
909         l) LIBSRC="$OPTARG";;
910         D) INSTALL_LIB="$OPTARG";;
911         d) INSTALL_DIR="$OPTARG/$TRAILER";;
912         G) GLOM=yes; GLOMNAME="$OPTARG";;
913         P|X|x) echo "-$opt is obsolete; ignored";;
914         h) usage "${INSTALL}: installs unix and modules";;
915         R) x=$OPTARG; restore_state; OPTIND=$x;;
916         i) IMPL="$OPTARG";;
917         o) OBJD="$OPTARG";;
918         K) KMDB="no";;
919         3) WANT64="no";;
920         6) WANT32="no";;
921         \?) usage "Illegal option";;
922     esac
923 done
924 shift `expr $OPTIND - 1`

```

```

926 ENV_NAME=`basename $ENV_PATH`

```

```

928 #
929 # The rest of the command line is a list of individual files to copy.
930 # If non-null, this list overrides the -uma options.
931 #

```

```

933 if [[ $# -gt 0 ]]; then
934     files="$*"

```

3

new/usr/src/tools/scripts/Install.sh

```

935         KMDB="no"
936 fi

938 case "$VERBOSE" in
939     v) V="v"; SHV="x";;
940     V) V="v"; SHV="x"; set -x;;
941     q) V=""; SHV="";;
942 esac

```

```

944 #
945 # Create temp directory for Install's files
946 #

```

```

948 tstmkdir $INSTALL_DIR

```

```

950 TARFILE=$INSTALL_DIR/Install.${KARCH}.tar
951 INSTALL_FILES=$INSTALL_DIR/$KARCH

```

```

953 #
954 # Extract the target machine and target directory from a target of the
955 # form [user@]machine:/dir .
956 #

```

```

958 if [ "$IMODE" != "n" ]; then
959     eval `echo $TARGET | /usr/xpg4/bin/awk -F: '{
959         eval `echo $TARGET | nawk -F: '{
960             if (NF != 2 || !length($1) || !length($2))
961                 print "usage \"Invalid target\""
962             m = $1; d = $2
963             if ($1 ~ /@/) {
964                 k = split($1, f, "@");
965                 if (k != 2 || !length(f[1]) || !length(f[2]))
966                     print "usage \"Invalid target\""
967                 u = f[1]; m = f[2]
968             }
969             print "TARGET_USER=" u ";";
970             print "TARGET_MACHINE=" m ";";
971             print "TARGET_DIR=" d ";";
972         }`
973     if [ -z "$TARGET_USER" ]; then
974         TARGET_USER=$LOGNAME
975     fi
976 fi

```

```

978 #
979 # Allow the use of library source or target for the install
980 #

```

```

982 if [ -n "$LIBSRC" ]; then
983     LIBSRC=`basename $LIBSRC .tar`.tar
984     TARFILE=$INSTALL_LIB/$LIBSRC
985     test -s $TARFILE || fail "Can't find tarfile $TARFILE"
986     verbose "Installing from library tarfile $TARFILE"
987     STATE=3
988 elif [ "$LIBCREATE" = "yes" ]; then
989     tstmkdir $INSTALL_LIB
990     TARFILE="$INSTALL_LIB/${ENV_NAME}.${KARCH}.tar"
991 fi

```

```

993 #
994 # The next few lines allow recovery and activation with -R,
995 # and library installs with -l.
996 #

```

```

998 [[ $STATE -eq 1 ]] && copy_kmdb
999 [[ $STATE -eq 2 ]] && make_tarfile

```

4

new/usr/src/tools/scripts/Install.sh

5

```
1000 [[ $STATE -eq 3 ]] && remote_install
1001 [[ $STATE -eq 4 ]] && okexit

1003 save_state

1005 cd $DOT
1006 DOTDOT='cd ../ pwd`

1008 #
1009 # Try to be smart: if DOTDOT ends in uts, then infer UTS and KARCH from DOT
1010 # Otherwise, if SRC is set, infer UTS = $SRC/uts.
1011 #

1013 if [ "`basename $DOTDOT`" = "uts" ]; then
1014     UTS=$DOTDOT
1015     KARCH=`basename $DOT`
1016     if [ ! -n "$SRC" ]; then
1017         SRC=`dirname $DOTDOT`
1018         verbose "Setting SRC to $SRC"
1019     fi
1020     export SRC
1021 fi

1023 if [ -z "$UTS" -a -n "$SRC" ]; then
1024     UTS="$SRC/uts"
1025     test -n "$KARCH" || fail "no karch specified (e.g. -k sun4u)"
1026 fi

1028 if [ "$LIBCREATE" = "yes" ]; then
1029     TARFILE=$INSTALL_LIB/${ENV_NAME}.${KARCH}.tar
1030 else
1031     TARFILE=$INSTALL_DIR/Install.${KARCH}.tar
1032 fi
1033 INSTALL_FILES=$INSTALL_DIR/$KARCH
1034 save_state

1036 cd $DOT
1037 test -z "$UTS" && fail 'Cannot find kernel sources -- $SRC not set'
1038 test -d "$UTS" || fail "${UTS}: no such directory"

1040 #
1041 # Convert UTS into an absolute path.
1042 #

1044 cd $UTS
1045 UTS=`pwd`

1047 test "`basename $UTS`" = "uts" || \
1048     verbose "Warning: source path $UTS doesn't end in 'uts'"

1050 remove_dir $INSTALL_DIR/$KARCH
1051 rm -f $TARFILE

1053 copy_kernel      # sets STATE=1 if successful
1054 copy_kmdb        # sets STATE=2 if successful
1055 make_tarfile     # sets STATE=3 if successful
1056 remote_install  # sets STATE=4 if successful

1058 okexit
```

new/usr/src/tools/scripts/bindrop.sh

1

```
*****
7705 Sun May 26 18:23:47 2013
new/usr/src/tools/scripts/bindrop.sh
XXX Remove awk(1)
*****
_____unchanged_portion_omitted_____

66 nondebug=n
67 while getopts n flag; do
68     case $flag in
69         n)
70             nondebug=y
71             ;;
72         ?)
73             print -u2 "usage: $usage"
74             exit 1
75             ;;
76         esac
77 done
78 shift=$((OPTIND - 1))

80 if [[ $# -ne 1 ]]; then
81     print -u2 "usage: $usage"
82     exit 1
83 fi

85 tarfile="$CODEMGR_WS/$1.$isa.tar"

87 rootdir="root_$isa"
88 suffix=""
89 if [[ "$nondebug" = y ]]; then
90     rootdir="root_$isa-nd"
91     suffix="-nd"
92 fi

94 [[ -d "${ROOT}${suffix}-closed" ]] || fail "can't find closed root proto area."

96 tmpdir=$(mktemp -dt bindropXXXXX)
97 [[ -n "$tmpdir" ]] || fail "can't create temporary directory."
98 mkdir -p "$tmpdir/closed/$rootdir" || exit 1

100 #
101 # This will hold a temp list of directories that must be kept, even if
102 # empty.
103 #
104 needdirs=$(mktemp -t needdirsXXXXX)
105 [[ -n "$needdirs" ]] || fail "can't create temporary directory list file."

107 #
108 # Copy the closed root parallel tree into a temp directory.
109 #
110 (cd "${ROOT}${suffix}-closed"; tar cf - .) | (cd "$tmpdir/closed/$rootdir"; tar

112 #
113 # Remove internal ON crypto signing certs
114 #
115 delete="
116     etc/certs/SUNWosnetSE
117     etc/certs/SUNWosnetSolaris
118     etc/crypto/certs/SUNWosnet
119     etc/crypto/certs/SUNWosnetLimited
120     etc/crypto/certs/SUNWosnetCF
121     etc/crypto/certs/SUNWosnetCFLimited
122     "

124 #
```

new/usr/src/tools/scripts/bindrop.sh

2

```
125 # Remove miscellaneous files that we don't want to ship.
126 #

128 # SUNWsvvs (SVVS test drivers).
129 delete="$delete
130     usr/include/sys/svvslo.h
131     usr/include/sys/tidg.h
132     usr/include/sys/tivc.h
133     usr/include/sys/tmux.h
134     usr/kernel/drv/amd64/svvslo
135     usr/kernel/drv/amd64/tidg
136     usr/kernel/drv/amd64/tivc
137     usr/kernel/drv/amd64/tmux
138     usr/kernel/drv/sparcv9/svvslo
139     usr/kernel/drv/sparcv9/tidg
140     usr/kernel/drv/sparcv9/tivc
141     usr/kernel/drv/sparcv9/tmux
142     usr/kernel/drv/svvslo
143     usr/kernel/drv/svvslo.conf
144     usr/kernel/drv/tidg
145     usr/kernel/drv/tidg.conf
146     usr/kernel/drv/tivc
147     usr/kernel/drv/tivc.conf
148     usr/kernel/drv/tmux
149     usr/kernel/drv/tmux.conf
150     usr/kernel/strmod/amd64/lmodb
151     usr/kernel/strmod/amd64/lmode
152     usr/kernel/strmod/amd64/lmodr
153     usr/kernel/strmod/amd64/lmodt
154     usr/kernel/strmod/lmodb
155     usr/kernel/strmod/lmode
156     usr/kernel/strmod/lmodr
157     usr/kernel/strmod/lmodt
158     usr/kernel/strmod/sparcv9/lmodb
159     usr/kernel/strmod/sparcv9/lmode
160     usr/kernel/strmod/sparcv9/lmodr
161     usr/kernel/strmod/sparcv9/lmodt
162 "
163 # encumbered binaries and associated files
164 delete="$delete
165     kernel/drv/amd64/bmc
166     kernel/drv/bmc
167     kernel/drv/bmc.conf
168     kernel/drv/ifp.conf
169     kernel/drv/sparcv9/ifp
170     kernel/drv/sparcv9/isp
171     kernel/drv/sparcv9/qus
172     kernel/drv/spwr
173     kernel/kmdb/sparcv9/isp
174     usr/has/bin/ksh
175     usr/has/bin/pfksh
176     usr/has/bin/rksh
177     usr/include/sys/scsi/adapters/ifpcmd.h
178     usr/include/sys/scsi/adapters/ifpio.h
179     usr/include/sys/scsi/adapters/ifpmail.h
180     usr/include/sys/scsi/adapters/ifpreg.h
181     usr/include/sys/scsi/adapters/ifpvar.h
182     usr/include/sys/scsi/adapters/ispcmd.h
183     usr/include/sys/scsi/adapters/ispmail.h
184     usr/include/sys/scsi/adapters/ispreg.h
185     usr/include/sys/scsi/adapters/ispvar.h
186     usr/lib/mdb/kvm/sparcv9/isp.so
187     usr/platform/sun4u/include/sys/memtestio.h
188     usr/platform/sun4u/include/sys/memtestio_ch.h
189     usr/platform/sun4u/include/sys/memtestio_chp.h
190     usr/platform/sun4u/include/sys/memtestio_ja.h
```

new/usr/src/tools/scripts/bindrop.sh

3

```
191     usr/platform/sun4u/include/sys/memtestio_jg.h
192     usr/platform/sun4u/include/sys/memtestio_sf.h
193     usr/platform/sun4u/include/sys/memtestio_sr.h
194     usr/platform/sun4u/include/sys/memtestio_u.h
195     usr/platform/sun4u/include/sys/memtestio_pn.h
196     usr/platform/sun4v/include/sys/memtestio.h
197     usr/platform/sun4v/include/sys/memtestio_ni.h
198     usr/platform/sun4v/include/sys/memtestio_v.h
199 "
200 # memory fault injector test framework
201 delete="$delete
202     usr/bin/mtst
203     platform/sun4u/kernel/drv/sparcv9/memtest
204     platform/sun4u/kernel/drv/memtest.conf
205     platform/sun4v/kernel/drv/sparcv9/memtest
206     platform/sun4v/kernel/drv/memtest.conf
207     kernel/drv/memtest.conf
208     kernel/drv/memtest
209     kernel/drv/amd64/memtest
210     usr/platform/i86pc/lib/mtst/mtst_AuthenticAMD_15.so
211     usr/platform/i86pc/lib/mtst/mtst_AuthenticAMD.so
212     usr/platform/i86pc/lib/mtst/mtst_generic.so
213     usr/platform/i86pc/lib/mtst/mtst_GenuineIntel.so
214 "
215 for f in $delete; do
216     rm -rf "$tmpdir/closed/$rootdir/$f"
217 done

219 #
220 # Remove any header files.  If they're in the closed tree, they're
221 # probably not freely redistributable.
222 #
223 (cd "$tmpdir/closed/$rootdir"; find . -name \*.h \
224     -a \! -name lc_core.h \
225     -a \! -name localedef.h \
226     -exec rm -f {} \;)

228 #
229 # Remove empty directories that the open tree doesn't need.
230 #
231 # Step 1: walk the redistributable manifests to find out which directories
232 # are specified in the open packages; save that list to a temporary
233 # file $needdirs.
234 #
235 MACH=$(uname -p)
236 (cd "$SRC/pkg/packages.$MACH"; \
237     /usr/xpg4/bin/awk '/^dir/ {sub(/.+ path=/, ""); print $1;}' *.metadata.*
238     nawk '/^dir/ {sub(/.+ path=/, ""); print $1;}' *.metadata.*.redist | \
239     sort -u > "$needdirs")

240 #
241 # Step 2: go to our closed directory, and find all the subdirectories,
242 # filtering out the ones needed by the open packages (saved in that
243 # temporary file).  Sort in reverse order, so that parent directories
244 # come after any subdirectories, and pipe that to rmdir.  If there are
245 # still any lingering files, rmdir will complain.  That's fine--we
246 # only want to delete empty directories--so redirect the complaints to
247 # /dev/null.
248 #
249 (cd "$tmpdir/closed/$rootdir"; \
250     find * -type d -print | /usr/xpg4/bin/grep -xv -f $needdirs | \
251     sort -r | xargs -l rmdir 2>/dev/null )

253 rm "$needdirs"

255 #
```

new/usr/src/tools/scripts/bindrop.sh

4

```
256 # Exclude signed crypto binaries; they are delivered in their
257 # own tarball.
258 #
259 ROOT="$tmpdir/closed/$rootdir" findcrypto "$SRC/tools/codesign/creds" |
260     awk '{ print $2 }' | (cd "$tmpdir/closed/$rootdir"; xargs rm -f)

262 #
263 # Add binary license files.
264 #

266 cp -p "$SRC/tools/opensolaris/BINARYLICENSE.txt" "$tmpdir/closed" || \
267     fail "can't add BINARYLICENSE.txt"
268 mkreadme "$tmpdir/closed"
269 if [ -f "$CODEMGR_WS/THIRDPARTYLICENSE.ON-BINARIES" ]; then
270     cp -p "$CODEMGR_WS/THIRDPARTYLICENSE.ON-BINARIES" "$tmpdir/closed"
271 fi

273 (cd "$tmpdir"; tar cf "$starfile" closed) || fail "can't create $starfile."
274 bzip2 -f "$starfile" || fail "can't compress $starfile".

276 rm -rf "$tmpdir"

278 exit 0
```

new/usr/src/tools/scripts/checkpaths.sh

1

```
*****
4524 Sun May 26 18:23:47 2013
new/usr/src/tools/scripts/checkpaths.sh
XXX Remove awk(1)
*****
1 #!/bin/ksh -p
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 #
24 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 #
28 # Quis custodiet ipsos custodiet?
29 #
30 if [ -z "$SRC" ]; then
31     SRC=$CODEMGR_WS/usr/src
32 fi
33 #
34 if [ -z "$CODEMGR_WS" -o ! -d "$CODEMGR_WS" -o ! -d "$SRC" ]; then
35     echo "$0: must be run from within a workspace."
36     exit 1
37 fi
38 #
39 cd $CODEMGR_WS || exit 1
40 #
41 # Use -b to tell this script to ignore derived (built) objects.
42 if [ "$1" = "-b" ]; then
43     b_flg=y
44 fi
45 #
46 # Not currently used; available for temporary workarounds.
47 args="-k NEVER_CHECK"
48 #
49 # We intentionally don't depend on $MACH here, and thus no $ROOT. If
50 # a proto area exists, then we use it. This allows this script to be
51 # run against gates (which should contain both SPARC and x86 proto
52 # areas), build workspaces (which should contain just one proto area),
53 # and unbuild workspaces (which contain no proto areas).
54 if [ "$b_flg" = y ]; then
55     rootlist=
56 elif [ $# -gt 0 ]; then
57     rootlist=$*
58 else
59     rootlist="$CODEMGR_WS/proto/root_sparc $CODEMGR_WS/proto/root_i386"
60 fi
```

new/usr/src/tools/scripts/checkpaths.sh

2

```
62 # If the closed source is not present, then exclude IKE from validation.
63 if [ "$CLOSED_IS_PRESENT" = no ]; then
64     excl="-e ^usr/include/ike/"
65 fi
66 #
67 for ROOT in $rootlist
68 do
69     case "$ROOT" in
70         *sparc|*sparc-nd)
71             arch=sparc
72             ;;
73         *i386|*i386-nd)
74             arch=i386
75             ;;
76         *)
77             echo "$ROOT has unknown architecture." >&2
78             exit 1
79             ;;
80     esac
81     if [ -d $ROOT ]; then
82         #
83         # This is the old-style packaging exception list, from
84         # the svr4-specific usr/src/pkgdefs
85         #
86         [ -f $SRC/pkgdefs/etc/exception_list_$arch ] && \
87             validate_paths '-s/\s*' $arch'$/' $excl -b $ROOT \
88                 $args $SRC/pkgdefs/etc/exception_list_$arch
89         #
90         # These are the new-style packaging exception lists,
91         # from the repository-wide exception_lists/ directory.
92         #
93         e="$CODEMGR_WS/exception_lists/packaging"
94         for f in $e; do
95             if [ -f $f ]; then
96                 /usr/xpg4/bin/awk 'NF == 1 || /[ ]\+ '$$arch'$/ { print; }'
97                 < $f | validate_paths -b $ROOT -n $f
98             fi
99         done
100     fi
101 done
102 #
103 # Two entries in the findunref exception_list deal with things created
104 # by nightly. Otherwise, this test could be run on an unmodified (and
105 # unbuild) workspace. We handle this by flagging the one that is
106 # present only on a built workspace (./*.out) and the one that's
107 # present only after a run of findunref (./*.ref) with ISUSED, and
108 # disabling all checks of them. The assumption is that the entries
109 # marked with ISUSED are always known to be good, thus the Latin quote
110 # at the top of the file.
111 #
112 # The exception_list is generated from whichever input files are appropriate
113 # for this workspace, so checking it obviates the need to check the inputs.
114 #
115 if [ -r $SRC/tools/findunref/exception_list ]; then
116     validate_paths -k ISUSED -r -e '\*' $SRC/tools/findunref/exception_list
117 fi
118 #
119 # These are straightforward.
120 if [ -d $SRC/xmod ]; then
121     # If the closed source is not present, then don't validate it.
122     if [ "$CLOSED_IS_PRESENT" = no ]; then
123         excl_cry="-e ^usr/closed"
124         excl_xmod="-e ^../closed"
125     fi
126     validate_paths $excl_cry $SRC/xmod/cry_files
```


new/usr/src/tools/scripts/checkpaths.sh

3

```
127     validate_paths $excl_xmod -b $SRC $SRC/xmod/xmod_files
128 fi

130 if [ -f $SRC/tools/opensolaris/license-list ]; then
131     excl=
132     if [ "$CLOSED_IS_PRESENT" = no ]; then
133         excl="-e ^usr/closed"
134     fi
135     sed -e 's/$.descrip/' < $SRC/tools/opensolaris/license-list | \
136         validate_paths -n SRC/tools/opensolaris/license-list $excl
137 fi

139 # Finally, make sure the that (req|inc).flg files are in good shape.
140 # If SCCS files are not expected to be present, though, then don't
141 # check them.
142 if [ ! -d "$CODEMGR_WS/Codemgr_wsdata" ]; then
143     f_flg='-f'
144 fi
145 # If the closed source is not present, then don't validate it.
146 if [ "$CLOSED_IS_PRESENT" = no ]; then
147     excl="-e ^usr/closed/"
148 fi

150 validate_flg $f_flg $excl

152 exit 0
```

new/usr/src/tools/scripts/nightly.sh

1

```
*****
87572 Sun May 26 18:23:48 2013
new/usr/src/tools/scripts/nightly.sh
XXX Remove awk(1)
*****
_____unchanged_portion_omitted_____

1821 trap cleanup 0
1822 trap cleanup_signal 1 2 3 15

1824 #
1825 # Generic lock file processing -- make sure that the lock file doesn't
1826 # exist. If it does, it should name the build host and PID. If it
1827 # doesn't, then make sure we can create it. Clean up locks that are
1828 # known to be stale (assumes host name is unique among build systems
1829 # for the workspace).
1830 #
1831 function create_lock {
1832     lockf=$1
1833     lockvar=$2

1835     ldir='dirname $lockf'
1836     [ -d $ldir ] || newdir $ldir || exit 1
1837     eval $lockvar=$lockf

1839     while ! staffer ln -s $hostname.$STAFFER.$$ $lockf 2> /dev/null; do
1840         basews='basename $CODEMGR_WS'
1841         ls -l $lockf | /usr/xpg4/bin/awk '{print $NF}' | IFS=. read host
1842         ls -l $lockf | awk '{print $NF}' | IFS=. read host user pid
1843         if [ "$host" != "$hostname" ]; then
1844             echo "$SMACH build of $basews apparently" \
1845                 "already started by $user on $host as $pid."
1846             exit 1
1847         elif kill -s 0 $pid 2>/dev/null; then
1848             echo "$SMACH build of $basews already started" \
1849                 "by $user as $pid."
1850             exit 1
1851         else
1852             # stale lock; clear it out and try again
1853             rm -f $lockf
1854         fi
1855     }
_____unchanged_portion_omitted_____

2432 type bringover_subversion > /dev/null 2>&1 || function bringover_subversion {
2433     typeset -x PATH=$PATH

2435     if [[ ! -d $CODEMGR_WS/.svn ]]; then
2436         staffer svn checkout $BRINGOVER_WS $CODEMGR_WS ||
2437             touch $TMPDIR/bringover_failed
2438     else
2439         typeset root
2440         root=$(staffer svn info $CODEMGR_WS |
2441             /usr/xpg4/bin/awk '/^Repository Root:/ {print $NF}')
2442         root=$(staffer svn info $CODEMGR_WS |
2443             /usr/xpg4/bin/awk '/^Repository Root:/ {print $NF}')
2444         if [[ $root != $BRINGOVER_WS ]]; then
2445             # We fail here because there's no way to update
2446             # from a named repo.
2447             cat <<-EOF
2448             \ $BRINGOVER_WS doesn't match repository root:
2449             \ $BRINGOVER_WS: $BRINGOVER_WS
2450             \ Repository root: $root
2451             EOF
2452             touch $TMPDIR/bringover_failed
2453         else
2454             # If a conflict happens, svn still exits 0.
2455             staffer svn update $CODEMGR_WS | tee $TMPDIR/pull.out ||
2456                 touch $TMPDIR/bringover_failed
2457             if grep "^C" $TMPDIR/pull.out > /dev/null 2>&1; then
2458                 touch $TMPDIR/bringover_failed
2459             fi
2460         fi
_____unchanged_portion_omitted_____

2467 #
2468 # Decide whether to bringover to the codemgr workspace
2469 #
2470 if [ "$n_FLAG" = "n" ]; then
2471     PARENT_SCM_TYPE=$(parent_wstype)

2473     if [[ $SCM_TYPE != none && $SCM_TYPE != $PARENT_SCM_TYPE ]]; then
2474         echo "cannot bringover from $PARENT_SCM_TYPE to $SCM_TYPE, " \
2475             "quitting at 'date.'" | tee -a $mail_msg_file >> $LOGFILE
2476     fi
2477     exit 1

2479     run_hook PRE_BRINGOVER

2481     echo "\n==== bringover to $CODEMGR_WS at 'date' ==== \n" >> $LOGFILE
2482     echo "\n==== BRINGOVER LOG ==== \n" >> $mail_msg_file

2484     eval "bringover_${PARENT_SCM_TYPE}" 2>&1 |
2485     tee -a $mail_msg_file >> $LOGFILE

2487     if [ -f $TMPDIR/bringover_failed ]; then
2488         rm -f $TMPDIR/bringover_failed
2489         build_ok=n
2490         echo "trouble with bringover, quitting at 'date.'" |
2491         tee -a $mail_msg_file >> $LOGFILE
2492     fi
2493     exit 1

2495     fi

2496     # It's possible that we used the bringover above to create
2497     # $CODEMGR_WS. If so, then SCM_TYPE was previously "none,"
2498     # but should now be the same as $BRINGOVER_WS.
2499     #
2500     [[ $SCM_TYPE = none ]] && SCM_TYPE=$PARENT_SCM_TYPE

2502     run_hook POST_BRINGOVER

2504     #
2505     # Possible transition from pre-split workspace to split
2506     # workspace. See if the bringover changed anything.
2507     #
2508     CLOSED_IS_PRESENT="$orig_closed_is_present"
2509     check_closed_tree

2511 else
2512     echo "\n==== No bringover to $CODEMGR_WS ==== \n" >> $LOGFILE
2513 fi

2515 if [[ "$O_FLAG" = y && "$CLOSED_IS_PRESENT" != "yes" ]]; then
2516     build_ok=n
2517     echo "OpenSolaris binary deliverables need usr/closed." \
2518         | tee -a "$mail_msg_file" >> $LOGFILE
2519     exit 1
2520 fi
```

new/usr/src/tools/scripts/nightly.sh

2

```
*****
87572 Sun May 26 18:23:48 2013
new/usr/src/tools/scripts/nightly.sh
XXX Remove awk(1)
*****
_____unchanged_portion_omitted_____

1821 trap cleanup 0
1822 trap cleanup_signal 1 2 3 15

1824 #
1825 # Generic lock file processing -- make sure that the lock file doesn't
1826 # exist. If it does, it should name the build host and PID. If it
1827 # doesn't, then make sure we can create it. Clean up locks that are
1828 # known to be stale (assumes host name is unique among build systems
1829 # for the workspace).
1830 #
1831 function create_lock {
1832     lockf=$1
1833     lockvar=$2

1835     ldir='dirname $lockf'
1836     [ -d $ldir ] || newdir $ldir || exit 1
1837     eval $lockvar=$lockf

1839     while ! staffer ln -s $hostname.$STAFFER.$$ $lockf 2> /dev/null; do
1840         basews='basename $CODEMGR_WS'
1841         ls -l $lockf | /usr/xpg4/bin/awk '{print $NF}' | IFS=. read host
1842         ls -l $lockf | awk '{print $NF}' | IFS=. read host user pid
1843         if [ "$host" != "$hostname" ]; then
1844             echo "$SMACH build of $basews apparently" \
1845                 "already started by $user on $host as $pid."
1846             exit 1
1847         elif kill -s 0 $pid 2>/dev/null; then
1848             echo "$SMACH build of $basews already started" \
1849                 "by $user as $pid."
1850             exit 1
1851         else
1852             # stale lock; clear it out and try again
1853             rm -f $lockf
1854         fi
1855     }
_____unchanged_portion_omitted_____

2432 type bringover_subversion > /dev/null 2>&1 || function bringover_subversion {
2433     typeset -x PATH=$PATH

2435     if [[ ! -d $CODEMGR_WS/.svn ]]; then
2436         staffer svn checkout $BRINGOVER_WS $CODEMGR_WS ||
2437             touch $TMPDIR/bringover_failed
2438     else
2439         typeset root
2440         root=$(staffer svn info $CODEMGR_WS |
2441             /usr/xpg4/bin/awk '/^Repository Root:/ {print $NF}')
2442         root=$(staffer svn info $CODEMGR_WS |
2443             /usr/xpg4/bin/awk '/^Repository Root:/ {print $NF}')
2444         if [[ $root != $BRINGOVER_WS ]]; then
2445             # We fail here because there's no way to update
2446             # from a named repo.
2447             cat <<-EOF
2448             \ $BRINGOVER_WS doesn't match repository root:
2449             \ $BRINGOVER_WS: $BRINGOVER_WS
2450             \ Repository root: $root
2451             EOF
2452             touch $TMPDIR/bringover_failed
2453         else
2454             # If a conflict happens, svn still exits 0.
2455             staffer svn update $CODEMGR_WS | tee $TMPDIR/pull.out ||
2456                 touch $TMPDIR/bringover_failed
2457             if grep "^C" $TMPDIR/pull.out > /dev/null 2>&1; then
2458                 touch $TMPDIR/bringover_failed
2459             fi
2460         fi
_____unchanged_portion_omitted_____

2467 #
2468 # Decide whether to bringover to the codemgr workspace
2469 #
2470 if [ "$n_FLAG" = "n" ]; then
2471     PARENT_SCM_TYPE=$(parent_wstype)

2473     if [[ $SCM_TYPE != none && $SCM_TYPE != $PARENT_SCM_TYPE ]]; then
2474         echo "cannot bringover from $PARENT_SCM_TYPE to $SCM_TYPE, " \
2475             "quitting at 'date.'" | tee -a $mail_msg_file >> $LOGFILE
2476     fi
2477     exit 1

2479     run_hook PRE_BRINGOVER

2481     echo "\n==== bringover to $CODEMGR_WS at 'date' ==== \n" >> $LOGFILE
2482     echo "\n==== BRINGOVER LOG ==== \n" >> $mail_msg_file

2484     eval "bringover_${PARENT_SCM_TYPE}" 2>&1 |
2485     tee -a $mail_msg_file >> $LOGFILE

2487     if [ -f $TMPDIR/bringover_failed ]; then
2488         rm -f $TMPDIR/bringover_failed
2489         build_ok=n
2490         echo "trouble with bringover, quitting at 'date.'" |
2491         tee -a $mail_msg_file >> $LOGFILE
2492     fi
2493     exit 1

2495     fi

2496     # It's possible that we used the bringover above to create
2497     # $CODEMGR_WS. If so, then SCM_TYPE was previously "none,"
2498     # but should now be the same as $BRINGOVER_WS.
2499     #
2500     [[ $SCM_TYPE = none ]] && SCM_TYPE=$PARENT_SCM_TYPE

2502     run_hook POST_BRINGOVER

2504     #
2505     # Possible transition from pre-split workspace to split
2506     # workspace. See if the bringover changed anything.
2507     #
2508     CLOSED_IS_PRESENT="$orig_closed_is_present"
2509     check_closed_tree

2511 else
2512     echo "\n==== No bringover to $CODEMGR_WS ==== \n" >> $LOGFILE
2513 fi

2515 if [[ "$O_FLAG" = y && "$CLOSED_IS_PRESENT" != "yes" ]]; then
2516     build_ok=n
2517     echo "OpenSolaris binary deliverables need usr/closed." \
2518         | tee -a "$mail_msg_file" >> $LOGFILE
2519     exit 1
2520 fi
```

```

2522 # Safeguards
2523 [[ -v CODEMGR_WS ]] || fatal_error "Error: Variable CODEMGR_WS not set."
2524 [[ -d "${CODEMGR_WS}" ]] || fatal_error "Error: ${CODEMGR_WS} is not a directory
2525 [[ -f "${CODEMGR_WS}/usr/src/Makefile" ]] || fatal_error "Error: ${CODEMGR_WS}/u

2527 echo "\n==== Build environment ====\n" | tee -a $build_envIRON_file >> $LOGFILE

2529 # System
2530 whence uname | tee -a $build_envIRON_file >> $LOGFILE
2531 uname -a 2>&1 | tee -a $build_envIRON_file >> $LOGFILE
2532 echo | tee -a $build_envIRON_file >> $LOGFILE

2534 # make
2535 whence $MAKE | tee -a $build_envIRON_file >> $LOGFILE
2536 $MAKE -v | tee -a $build_envIRON_file >> $LOGFILE
2537 echo "number of concurrent jobs = $DMAKE_MAX_JOBS" |
2538 tee -a $build_envIRON_file >> $LOGFILE

2540 #
2541 # Report the compiler versions.
2542 #

2544 if [[ ! -f $SRC/Makefile ]]; then
2545     build_ok=n
2546     echo "\nUnable to find \"Makefile\" in $SRC." | \
2547     tee -a $build_envIRON_file >> $LOGFILE
2548     exit 1
2549 fi

2551 ( cd $SRC
2552   for target in cc-version cc64-version java-version; do
2553     echo
2554     #
2555     # Put statefile somewhere we know we can write to rather than trip
2556     # over a read-only $srcroot.
2557     #
2558     rm -f $TMPDIR/make-state
2559     export SRC
2560     if $MAKE -K $TMPDIR/make-state -e $target 2>/dev/null; then
2561         continue
2562     fi
2563     touch $TMPDIR/nocompiler
2564     done
2565     echo
2566 ) | tee -a $build_envIRON_file >> $LOGFILE

2568 if [ -f $TMPDIR/nocompiler ]; then
2569     rm -f $TMPDIR/nocompiler
2570     build_ok=n
2571     echo "Aborting due to missing compiler." |
2572     tee -a $build_envIRON_file >> $LOGFILE
2573     exit 1
2574 fi

2576 # as
2577 whence as | tee -a $build_envIRON_file >> $LOGFILE
2578 as -V 2>&1 | head -1 | tee -a $build_envIRON_file >> $LOGFILE
2579 echo | tee -a $build_envIRON_file >> $LOGFILE

2581 # Check that we're running a capable link-editor
2582 whence ld | tee -a $build_envIRON_file >> $LOGFILE
2583 LDVER=`ld -V 2>&1`
2584 echo $LDVER | tee -a $build_envIRON_file >> $LOGFILE
2585 LDVER=`echo $LDVER | sed -e "s/.*-1\\.\\([0-9]*\\)\\.*/\\1/"`
2586 if [ `expr $LDVER \\< 422` -eq 1 ]; then
2587     echo "The link-editor needs to be at version 422 or higher to build" | \

```

```

2588     tee -a $build_envIRON_file >> $LOGFILE
2589     echo "the latest stuff. Hope your build works." | \
2590     tee -a $build_envIRON_file >> $LOGFILE
2591 fi

2593 #
2594 # Build and use the workspace's tools if requested
2595 #
2596 if [[ "$t_FLAG" = "y" || "$O_FLAG" = y ]]; then
2597     set_non_debug_build_flags

2599     build_tools ${TOOLS_PROTO}
2600     if [[ $? != 0 && "$t_FLAG" = y ]]; then
2601         use_tools $TOOLS_PROTO
2602     fi
2603 fi

2605 #
2606 # copy ihv proto area in addition to the build itself
2607 #
2608 if [ "$X_FLAG" = "y" ]; then
2609     copy_ihv_proto
2610 fi

2612 if [ "$i_FLAG" = "y" -a "$SH_FLAG" = "y" ]; then
2613     echo "\n==== NOT Building base OS-Net source ====\n" | \
2614     tee -a $LOGFILE >> $mail_msg_file
2615 else
2616     # timestamp the start of the normal build; the findunref tool uses it.
2617     touch $SRC/.build.tstamp

2619     normal_build
2620 fi

2622 #
2623 # Generate the THIRDPARTYLICENSE files if needed. This is done after
2624 # the build, so that dynamically-created license files are there.
2625 # It's done before findunref to help identify license files that need
2626 # to be added to tools/opensolaris/license-list.
2627 #
2628 if [ "$O_FLAG" = y -a "$build_ok" = y ]; then
2629     echo "\n==== Generating THIRDPARTYLICENSE files ====\n" |
2630     tee -a "$mail_msg_file" >> "$LOGFILE"

2632     if [ -d $ROOT/licenses/usr ]; then
2633         ( cd $ROOT/licenses ; \
2634           mktpl $SRC/pkg/license-list ) >> "$LOGFILE" 2>&1
2635         if (( $? != 0 )) ; then
2636             echo "Couldn't create THIRDPARTYLICENSE files" |
2637             tee -a "$mail_msg_file" >> "$LOGFILE"
2638         fi
2639     else
2640         echo "No licenses found under $ROOT/licenses" |
2641         tee -a "$mail_msg_file" >> "$LOGFILE"
2642     fi
2643 fi

2645 ORIG_SRC=$SRC
2646 BINARCHIVE=${CODEMGR_WS}/bin-${MACH}.cpio.Z

2648 if [ "$SE_FLAG" = "y" -o "$SD_FLAG" = "y" -o "$SH_FLAG" = "y" ]; then
2649     save_binaries
2650 fi

2653 # EXPORT_SRC comes after CRYPT_SRC since a domestic build will need

```

new/usr/src/tools/scripts/nightly.sh

5

```

2654 # $SRC pointing to the export_source usr/src.

2656 if [ "$SE_FLAG" = "y" -o "$SD_FLAG" = "y" -o "$SH_FLAG" = "y" ]; then
2657     if [ "$SD_FLAG" = "y" -a $build_ok = y ]; then
2658         set_up_source_build ${CODEMGR_WS} ${CRYPT_SRC} CRYPT_SRC
2659     fi

2661     if [ $build_ok = y ]; then
2662         set_up_source_build ${CODEMGR_WS} ${EXPORT_SRC} EXPORT_SRC
2663     fi
2664 fi

2666 if [ "$SD_FLAG" = "y" -a $build_ok = y ]; then
2667     # drop the crypt files in place.
2668     cd ${EXPORT_SRC}
2669     echo "\nexttracting crypt_files.cpio.Z onto export_source.\n" \
2670     >> ${LOGFILE}
2671     zcat ${CODEMGR_WS}/crypt_files.cpio.Z | \
2672     cpio -idmucvB 2>/dev/null >> ${LOGFILE}
2673     if [ "$?" = "0" ]; then
2674         echo "\n=== DOMESTIC extraction succeeded ===\n" \
2675         >> $mail_msg_file
2676     else
2677         echo "\n=== DOMESTIC extraction failed ===\n" \
2678         >> $mail_msg_file
2679     fi

2681 fi

2683 if [ "$SO_FLAG" = "y" -a "$build_ok" = y ]; then
2684     #
2685     # Copy the open sources into their own tree.
2686     # If copy_source fails, it will have already generated an
2687     # error message and set build_ok=n, so we don't need to worry
2688     # about that here.
2689     #
2690     copy_source $CODEMGR_WS $OPEN_SRCDIR OPEN_SOURCE usr/src
2691 fi

2693 if [ "$SO_FLAG" = "y" -a "$build_ok" = y ]; then
2694     SRC=$OPEN_SRCDIR/usr/src
2695     export CLOSED_IS_PRESENT=no
2696 fi

2698 if is_source_build && [ $build_ok = y ]; then
2699     # remove proto area(s) here, since we don't clobber
2700     rm -rf `allprotos`
2701     if [ "$t_FLAG" = "y" ]; then
2702         set_non_debug_build_flags
2703         ORIG_TOOLS=$TOOLS
2704         #
2705         # SRC was set earlier to point to the source build
2706         # source tree (e.g., $EXPORT_SRC).
2707         #
2708         TOOLS=${SRC}/tools
2709         TOOLS_PROTO=${TOOLS}/${TOOLS_PROTO_REL}; export TOOLS_PROTO
2710         build_tools ${TOOLS_PROTO}
2711         if [[ $? != 0 ]]; then
2712             use_tools ${TOOLS_PROTO}
2713         fi
2714     fi

2716     export EXPORT_RELEASE_BUILD ; EXPORT_RELEASE_BUILD=#
2717     normal_build
2718 fi

```

new/usr/src/tools/scripts/nightly.sh

6

```

2720 #
2721 # There are several checks that need to look at the proto area, but
2722 # they only need to look at one, and they don't care whether it's
2723 # DEBUG or non-DEBUG.
2724 #
2725 if [[ "$MULTI_PROTO" = yes && "$D_FLAG" = n ]]; then
2726     checkroot=$ROOT-nd
2727 else
2728     checkroot=$ROOT
2729 fi

2731 if [ "$build_ok" = "y" ]; then
2732     echo "\n=== Creating protolist system file at `date` ===" \
2733     >> $LOGFILE
2734     protolist $checkroot > $ATLOG/proto_list_${MACH}
2735     echo "=== protolist system file created at `date` ===\n" \
2736     >> $LOGFILE

2738     if [ "$N_FLAG" != "y" ]; then

2740         E1=
2741         f1=
2742         if [ -d "$SRC/pkgdefs" ]; then
2743             f1="$SRC/pkgdefs/etc/exception_list_${MACH}"
2744             if [ "$X_FLAG" = "y" ]; then
2745                 f1="$f1 $IA32_IHV_WS/usr/src/pkgdefs/etc/excepti
2746             fi
2747         fi

2749         for f in $f1; do
2750             if [ -f "$f" ]; then
2751                 E1="$E1 -e $f"
2752             fi
2753         done

2755         E2=
2756         f2=
2757         if [ -d "$SRC/pkg" ]; then
2758             f2="$f2 exceptions/packaging"
2759         fi

2761         for f in $f2; do
2762             if [ -f "$f" ]; then
2763                 E2="$E2 -e $f"
2764             fi
2765         done

2767         if [ -f "$REF_PROTO_LIST" ]; then
2768             #
2769             # For builds that copy the IHV proto area (-X), add the
2770             # IHV proto list to the reference list if the reference
2771             # was built without -X.
2772             #
2773             # For builds that don't copy the IHV proto area, add the
2774             # IHV proto list to the build's proto list if the
2775             # reference was built with -X.
2776             #
2777             # Use the presence of the first file entry of the cached
2778             # IHV proto list in the reference list to determine
2779             # whether it was built with -X or not.
2780             #
2781             IHV_REF_PROTO_LIST=$SRC/pkg/proto_list_ihv_${MACH}
2782             grepfor=$(/usr/xpg4/bin/awk ' $1 == "f" { print $2; exit }' \
2783             grepfor=$(awk ' $1 == "f" { print $2; exit }' \
2784             $IHV_REF_PROTO_LIST 2> /dev/null)

2784             if [ $? = 0 -a -n "$grepfor" ]; then

```

```

2785         if [ "$X_FLAG" = "y" ]; then
2786             grep -w "$grepfor" \
2787                 $REF_PROTO_LIST > /dev/null
2788             if [ ! "$?" = "0" ]; then
2789                 REF_IHV_PROTO="-d $IHV_REF_PROTO
2790             fi
2791         else
2792             grep -w "$grepfor" \
2793                 $REF_PROTO_LIST > /dev/null
2794             if [ "$?" = "0" ]; then
2795                 IHV_PROTO_LIST="$IHV_REF_PROTO_L
2796             fi
2797         fi
2798     fi
2799 fi
2800
2802 if [ "$N_FLAG" != "y" -a -f $SRC/pkgdefs/Makefile ]; then
2803     echo "\n=== Impact on SVr4 packages ===\n" >> $mail_msg_file
2804     #
2805     # Compare the build's proto list with current package
2806     # definitions to audit the quality of package
2807     # definitions and makefile install targets. Use the
2808     # current exception list.
2809     #
2810     PKGDEFS_LIST=""
2811     for d in $abssrkdirs; do
2812         if [ -d $d/pkgdefs ]; then
2813             PKGDEFS_LIST="$PKGDEFS_LIST -d $d/pkgdefs"
2814         fi
2815     done
2816     if [ "$X_FLAG" = "y" -a \
2817         -d $IA32_IHV_WS/usr/src/pkgdefs ]; then
2818         PKGDEFS_LIST="$PKGDEFS_LIST -d $IA32_IHV_WS/usr/src/pkgd
2819     fi
2820     $PROTOCMPTRSE \
2821         "Files missing from the proto area:" \
2822         "Files missing from packages:" \
2823         "Inconsistencies between pkgdefs and proto area:" \
2824         ${E1} \
2825         ${PKGDEFS_LIST} \
2826         $ATLOG/proto_list_${MACH} \
2827         >> $mail_msg_file
2828 fi
2830
2831 if [ "$N_FLAG" != "y" -a -d $SRC/pkg ]; then
2832     echo "\n=== Validating manifests against proto area ===\n" \
2833         >> $mail_msg_file
2834     ( cd $SRC/pkg ; $MAKE -e protocmp ROOT="$checkroot" ) \
2835         >> $mail_msg_file
2836 fi
2838
2839 if [ "$N_FLAG" != "y" -a -f "$REF_PROTO_LIST" ]; then
2840     echo "\n=== Impact on proto area ===\n" >> $mail_msg_file
2841     if [ -n "$E2" ]; then
2842         ELIST=$E2
2843     else
2844         ELIST=$E1
2845     fi
2846     $PROTOCMPTRSE \
2847         "Files in yesterday's proto area, but not today's:" \
2848         "Files in today's proto area, but not yesterday's:" \
2849         "Files that changed between yesterday and today:" \
2850         ${ELIST} \
2851         -d $REF_PROTO_LIST \

```

```

2851         $REF_IHV_PROTO \
2852         $ATLOG/proto_list_${MACH} \
2853         $IHV_PROTO_LIST \
2854         >> $mail_msg_file
2855     fi
2856 fi
2858
2859 if [ "$u_FLAG" = "y" -a "$build_ok" = "y" ]; then
2860     staffer cp $ATLOG/proto_list_${MACH} \
2861         $PARENT_WS/usr/src/proto_list_${MACH}
2862 fi
2863
2864 # Update parent proto area if necessary. This is done now
2865 # so that the proto area has either DEBUG or non-DEBUG kernels.
2866 # Note that this clears out the lock file, so we can dispense with
2867 # the variable now.
2868 if [ "$U_FLAG" = "y" -a "$build_ok" = "y" ]; then
2869     echo "\n=== Copying proto area to $NIGHTLY_PARENT_ROOT ===\n" | \
2870         tee -a $LOGFILE >> $mail_msg_file
2871     rm -rf $NIGHTLY_PARENT_ROOT/*
2872     unset Ulockfile
2873     mkdir -p $NIGHTLY_PARENT_ROOT
2874     if [[ "$MULTI_PROTO" = no || "$D_FLAG" = y ]]; then
2875         ( cd $ROOT; tar cf - . |
2876             ( cd $NIGHTLY_PARENT_ROOT; umask 0; tar xpf - ) ) 2>&1 |
2877             tee -a $mail_msg_file >> $LOGFILE
2878     fi
2879     if [[ "$MULTI_PROTO" = yes && "$F_FLAG" = n ]]; then
2880         rm -rf $NIGHTLY_PARENT_ROOT-nd/*
2881         mkdir -p $NIGHTLY_PARENT_ROOT-nd
2882         cd $ROOT-nd
2883         ( tar cf - . |
2884             ( cd $NIGHTLY_PARENT_ROOT-nd; umask 0; tar xpf - ) ) 2>&1 |
2885             tee -a $mail_msg_file >> $LOGFILE
2886     fi
2887     if [ -n "${NIGHTLY_PARENT_TOOLS_ROOT}" ]; then
2888         echo "\n=== Copying tools proto area to $NIGHTLY_PARENT_TOOLS_R
2889             tee -a $LOGFILE >> $mail_msg_file
2890         rm -rf $NIGHTLY_PARENT_TOOLS_ROOT/*
2891         mkdir -p $NIGHTLY_PARENT_TOOLS_ROOT
2892         if [[ "$MULTI_PROTO" = no || "$D_FLAG" = y ]]; then
2893             ( cd $TOOLS_PROTO; tar cf - . |
2894                 ( cd $NIGHTLY_PARENT_TOOLS_ROOT;
2895                     umask 0; tar xpf - ) ) 2>&1 |
2896                 tee -a $mail_msg_file >> $LOGFILE
2897         fi
2898     fi
2899 fi
2900 #
2901 # ELF verification: ABI (-A) and runtime (-r) checks
2902 #
2903 if [[ ($build_ok = y) && ( ($A_FLAG = y) || ($r_FLAG = y) ) ]]; then
2904     # Directory ELF-data.$MACH holds the files produced by these tests.
2905     elf_ddir=$SRC/ELF-data.$MACH
2907
2908 # If there is a previous ELF-data backup directory, remove it. Then,
2909 # rotate current ELF-data directory into its place and create a new
2910 # empty directory
2911 rm -rf $elf_ddir.ref
2912 if [[ -d $elf_ddir ]]; then
2913     mv $elf_ddir $elf_ddir.ref
2914 fi
2915 mkdir -p $elf_ddir
2916
2917 # Call find_elf to produce a list of the ELF objects in the proto area.

```

```

2917 # This list is passed to check_rtime and interface_check, preventing
2918 # them from separately calling find_elf to do the same work twice.
2919 find_elf -fr $checkroot > $self_ddir/object_list

2921 if [[ $A_FLAG = y ]]; then
2922     echo "\n=== Check versioning and ABI information ===\n" | \
2923     tee -a $LOGFILE >> $mail_msg_file

2925 # Produce interface description for the proto. Report errors.
2926 interface_check -o -w $self_ddir -f object_list \
2927     -i interface -E interface.err
2928 if [[ -s $self_ddir/interface.err ]]; then
2929     tee -a $LOGFILE < $self_ddir/interface.err \
2930     >> $mail_msg_file
2931 fi

2933 # If ELF_DATA_BASELINE_DIR is defined, compare the new interface
2934 # description file to that from the baseline gate. Issue a
2935 # warning if the baseline is not present, and keep going.
2936 if [[ "$SELF_DATA_BASELINE_DIR" != '' ]]; then
2937     base_ifile="$SELF_DATA_BASELINE_DIR/interface"

2939     echo "\n=== Compare versioning and ABI information" \
2940     "to baseline ===\n" | \
2941     tee -a $LOGFILE >> $mail_msg_file
2942     echo "Baseline: $base_ifile\n" >> $LOGFILE

2944     if [[ -f $base_ifile ]]; then
2945         interface_cmp -d -o $base_ifile \
2946             $self_ddir/interface > $self_ddir/interface.cm
2947         if [[ -s $self_ddir/interface.cm ]]; then
2948             echo | tee -a $LOGFILE >> $mail_msg_file
2949             tee -a $LOGFILE < \
2950                 $self_ddir/interface.cm \
2951                 >> $mail_msg_file
2952         fi
2953     else
2954         echo "baseline not available. comparison" \
2955         "skipped" | \
2956         tee -a $LOGFILE >> $mail_msg_file
2957     fi

2959 fi
2960 fi

2962 if [[ $r_FLAG = y ]]; then
2963     echo "\n=== Check ELF runtime attributes ===\n" | \
2964     tee -a $LOGFILE >> $mail_msg_file

2966 # If we're doing a DEBUG build the proto area will be left
2967 # with debuggable objects, thus don't assert -s.
2968 if [[ $D_FLAG = y ]]; then
2969     rtime_sflag=""
2970 else
2971     rtime_sflag="-s"
2972 fi
2973 check_rtime -i -m -v $rtime_sflag -o -w $self_ddir \
2974     -D object_list -f object_list -E runtime.err \
2975     -I runtime.attr.raw

2977 # check_rtime -I output needs to be sorted in order to
2978 # compare it to that from previous builds.
2979 sort $self_ddir/runtime.attr.raw > $self_ddir/runtime.attr
2980 rm $self_ddir/runtime.attr.raw

2982 # Report errors

```

```

2983 if [[ -s $self_ddir/runtime.err ]]; then
2984     tee -a $LOGFILE < $self_ddir/runtime.err \
2985     >> $mail_msg_file
2986 fi

2988 # If there is an ELF-data directory from a previous build,
2989 # then diff the attr files. These files contain information
2990 # about dependencies, versioning, and runpaths. There is some
2991 # overlap with the ABI checking done above, but this also
2992 # flushes out non-ABI interface differences along with the
2993 # other information.
2994 echo "\n=== Diff ELF runtime attributes" \
2995     "(since last build) ===\n" | \
2996     tee -a $LOGFILE >> $mail_msg_file

2998 if [[ -f $self_ddir.ref/runtime.attr ]]; then
2999     diff $self_ddir.ref/runtime.attr \
3000         $self_ddir/runtime.attr \
3001         >> $mail_msg_file
3002 fi
3003 fi

3005 # If -u set, copy contents of ELF-data.$MACH to the parent workspace.
3006 if [[ "$u_FLAG" = "y" ]]; then
3007     p_elf_ddir=$PARENT_WS/usr/src/ELF-data.$MACH

3009 # If parent lacks the ELF-data.$MACH directory, create it
3010 if [[ ! -d $p_elf_ddir ]]; then
3011     staffer mkdir -p $p_elf_ddir
3012 fi

3014 # These files are used asynchronously by other builds for ABI
3015 # verification, as above for the -A option. As such, we require
3016 # the file replacement to be atomic. Copy the data to a temp
3017 # file in the same filesystem and then rename into place.
3018 (
3019     cd $self_ddir
3020     for elf_dfile in *; do
3021         staffer cp $self_dfile \
3022             ${p_elf_ddir}/${elf_dfile}.new
3023         staffer mv -f ${p_elf_ddir}/${elf_dfile}.new \
3024             ${p_elf_ddir}/${elf_dfile}
3025     done
3026 )
3027 fi
3028 fi

3030 # DEBUG lint of kernel begins

3032 if [ "$i_CMD_LINE_FLAG" = "n" -a "$l_FLAG" = "y" ]; then
3033     if [ "$LINTDIRS" = "" ]; then
3034         # LINTDIRS="$SRC/uts y $SRC/stand y $SRC/psm y"
3035         LINTDIRS="$SRC y"
3036     fi
3037     set $LINTDIRS
3038     while [ $# -gt 0 ]; do
3039         dolint $1 $2; shift; shift
3040     done
3041 else
3042     echo "\n=== No '$MAKE lint' ===\n" >> $LOGFILE
3043 fi

3045 # "make check" begins

3047 if [ "$i_CMD_LINE_FLAG" = "n" -a "$C_FLAG" = "y" ]; then
3048     # remove old check.out

```

```

3049     rm -f $SRC/check.out
3051     rm -f $SRC/check-{$MACH}.out
3052     cd $SRC
3053     $MAKE -ek check ROOT="$checkroot" 2>&1 | tee -a $SRC/check-{$MACH}.out \
3054     >> $LOGFILE
3055     echo "\n==== cstyle/hdrchk errors ==== \n" >> $mail_msg_file

3057     grep ":" $SRC/check-{$MACH}.out |
3058     egrep -v "Ignoring unknown host" | \
3059     sort | uniq >> $mail_msg_file
3060 else
3061     echo "\n==== No '$MAKE check' ==== \n" >> $LOGFILE
3062 fi

3064 echo "\n==== Find core files ==== \n" | \
3065     tee -a $LOGFILE >> $mail_msg_file

3067 find $absrsrcdirs -name core -a -type f -exec file {} \; | \
3068     tee -a $LOGFILE >> $mail_msg_file

3070 if [ "$f_FLAG" = "y" -a "$build_ok" = "y" ]; then
3071     echo "\n==== Diff unreferenced files (since last build) ==== \n" \
3072     | tee -a $LOGFILE >> $mail_msg_file
3073     rm -f $SRC/unref-{$MACH}.ref
3074     if [ -f $SRC/unref-{$MACH}.out ]; then
3075         mv $SRC/unref-{$MACH}.out $SRC/unref-{$MACH}.ref
3076     fi

3078     findunref -S $SCM_TYPE -t $SRC/.build.tstamp -s usr $CODEMGR_WS \
3079     ${TOOLS}/findunref/exception_list 2>> $mail_msg_file | \
3080     sort > $SRC/unref-{$MACH}.out

3082     if [ ! -f $SRC/unref-{$MACH}.ref ]; then
3083         cp $SRC/unref-{$MACH}.out $SRC/unref-{$MACH}.ref
3084     fi

3086     diff $SRC/unref-{$MACH}.ref $SRC/unref-{$MACH}.out >> $mail_msg_file
3087 fi

3089 #
3090 # Generate the OpenSolaris deliverables if requested. Some of these
3091 # steps need to come after findunref and are commented below.
3092 #

3094 # If we are doing an OpenSolaris _source_build (-S 0) then we do
3095 # not have usr/closed available to us to generate closedbins from,
3096 # so skip this part.
3097 if [ "$SO_FLAG" = n -a "$O_FLAG" = y -a "$build_ok" = y ]; then
3098     echo "\n==== Generating OpenSolaris tarballs ==== \n" | \
3099     tee -a $mail_msg_file >> $LOGFILE

3101     cd $CODEMGR_WS

3103     #
3104     # This step grovels through the package manifests, so it
3105     # must come after findunref.
3106     #
3107     # We assume no DEBUG vs non-DEBUG package content variation
3108     # here; if that changes, then the "make all" in $SRC/pkg will
3109     # need to be moved into the conditionals and repeated for each
3110     # different build.
3111     #
3112     echo "Generating closed binaries tarball(s)..." >> $LOGFILE
3113     closed_basename=on-closed-bins
3114     if [ "$D_FLAG" = y ]; then

```

```

3115         bindrop "$closed_basename" >> "$LOGFILE" 2>&1
3116         if (( $? != 0 )) ; then
3117             echo "Couldn't create DEBUG closed binaries." |
3118             tee -a $mail_msg_file >> $LOGFILE
3119             build_ok=n
3120         fi
3121     fi
3122     if [ "$F_FLAG" = n ]; then
3123         bindrop -n "$closed_basename-nd" >> "$LOGFILE" 2>&1
3124         if (( $? != 0 )) ; then
3125             echo "Couldn't create non-DEBUG closed binaries." |
3126             tee -a $mail_msg_file >> $LOGFILE
3127             build_ok=n
3128         fi
3129     fi

3131     echo "Generating README.opensolaris..." >> $LOGFILE
3132     cat $SRC/tools/opensolaris/README.opensolaris.tmpl | \
3133     mkreadme_osol $CODEMGR_WS/README.opensolaris >> $LOGFILE 2>&1
3134     if (( $? != 0 )) ; then
3135         echo "Couldn't create README.opensolaris." |
3136         tee -a $mail_msg_file >> $LOGFILE
3137         build_ok=n
3138     fi
3139 fi

3141 # Verify that the usual lists of files, such as exception lists,
3142 # contain only valid references to files. If the build has failed,
3143 # then don't check the proto area.
3144 CHECK_PATHS=${CHECK_PATHS:-y}
3145 if [ "$CHECK_PATHS" = y -a "$N_FLAG" != y ]; then
3146     echo "\n==== Check lists of files ==== \n" | tee -a $LOGFILE \
3147     >> $mail_msg_file
3148     arg=-b
3149     [ "$build_ok" = y ] && arg=
3150     checkpaths $arg $checkroot 2>&1 | tee -a $LOGFILE >> $mail_msg_file
3151 fi

3153 if [ "$M_FLAG" != "y" -a "$build_ok" = y ]; then
3154     echo "\n==== Impact on file permissions ==== \n" \
3155     >> $mail_msg_file

3157     abspkgdefs=
3158     abspkg=
3159     for d in $absrsrcdirs; do
3160         if [ -d "$d/pkgdefs" ]; then
3161             abspkgdefs="$abspkgdefs $d"
3162         fi
3163         if [ -d "$d/pkg" ]; then
3164             abspkg="$abspkg $d"
3165         fi
3166     done

3168     if [ -n "$abspkgdefs" ]; then
3169         pmodes -qvdp \
3170         'find $abspkgdefs -name pkginfo.tmpl -print -o \
3171         -name .del\* -prune | sed -e 's:/pkginfo.tmpl$::' | \
3172         sort -u' >> $mail_msg_file
3173     fi

3175     if [ -n "$abspkg" ]; then
3176         for d in "$abspkg"; do
3177             ( cd $d/pkg ; $MAKE -e pmodes ) >> $mail_msg_file
3178         done
3179     fi
3180 fi

```

```

3182 if [ "$w_FLAG" = "y" -a "$build_ok" = "y" ]; then
3183     if [[ "$MULTI_PROTO" = no || "$D_FLAG" = y ]]; then
3184         do_wsdiff DEBUG $ROOT.prev $ROOT
3185     fi
3187     if [[ "$MULTI_PROTO" = yes && "$F_FLAG" = n ]]; then
3188         do_wsdiff non-DEBUG $ROOT.nd.prev $ROOT.nd
3189     fi
3190 fi

3192 END_DATE=`date`
3193 echo "==== Nightly $maketype build completed: $END_DATE ==== | \
3194     tee -a $LOGFILE >> $build_time_file

3196 typeset -i10 hours
3197 typeset -Z2 minutes
3198 typeset -Z2 seconds

3200 elapsed_time=$SECONDS
3201 ((hours = elapsed_time / 3600 ))
3202 ((minutes = elapsed_time / 60 % 60))
3203 ((seconds = elapsed_time % 60))

3205 echo "\n==== Total build time ==== | \
3206     tee -a $LOGFILE >> $build_time_file
3207 echo "\nreal    ${hours}:${minutes}:${seconds}" | \
3208     tee -a $LOGFILE >> $build_time_file

3210 if [ "$u_FLAG" = "y" -a "$f_FLAG" = "y" -a "$build_ok" = "y" ]; then
3211     staffer cp ${SRC}/unref-${MACH}.out $PARENT_WS/usr/src/

3213     #
3214     # Produce a master list of unreferenced files -- ideally, we'd
3215     # generate the master just once after all of the nightlyies
3216     # have finished, but there's no simple way to know when that
3217     # will be. Instead, we assume that we're the last nightly to
3218     # finish and merge all of the unref-${MACH}.out files in
3219     # $PARENT_WS/usr/src/. If we are in fact the final ${MACH} to
3220     # finish, then this file will be the authoritative master
3221     # list. Otherwise, another ${MACH}'s nightly will eventually
3222     # overwrite ours with its own master, but in the meantime our
3223     # temporary "master" will be no worse than any older master
3224     # which was already on the parent.
3225     #

3227     set -- $PARENT_WS/usr/src/unref-*.out
3228     cp "$1" ${TMPDIR}/unref.merge
3229     shift

3231     for unref; do
3232         comm -12 ${TMPDIR}/unref.merge "$unref" > ${TMPDIR}/unref.$$
3233         mv ${TMPDIR}/unref.$$ ${TMPDIR}/unref.merge
3234     done

3236     staffer cp ${TMPDIR}/unref.merge $PARENT_WS/usr/src/unrefmaster.out
3237 fi

3239 #
3240 # All done save for the sweeping up.
3241 # (whichever exit we hit here will trigger the "cleanup" trap which
3242 # optionally sends mail on completion).
3243 #
3244 if [ "$build_ok" = "y" ]; then
3245     exit 0
3246 fi

```

```

3247 exit 1

```


new/usr/src/tools/scripts/protocmp.terse.sh

1

```
*****
1983 Sun May 26 18:23:51 2013
new/usr/src/tools/scripts/protocmp.terse.sh
XXX Remove nawk(1)
*****
1 #!/bin/ksh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License, Version 1.0 only
7 # (the "License"). You may not use this file except in compliance
8 # with the License.
9 #
10 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
11 # or http://www.opensolaris.org/os/licensing.
12 # See the License for the specific language governing permissions
13 # and limitations under the License.
14 #
15 # When distributing Covered Code, include this CDDL HEADER in each
16 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
17 # If applicable, add the following below this CDDL HEADER, with the
18 # fields enclosed by brackets "[]" replaced with your own identifying
19 # information: Portions Copyright [yyyy] [name of copyright owner]
20 #
21 # CDDL HEADER END
22 #
23 #
24 # Copyright 1993-2003 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 #ident "%Z%M% %I% %E% SMI"

28 opts=""
29 while getopts guplmsLv arg
30 do
31     if [ "$arg" = "?" ]; then
32         exit 1
33     fi
34     opts="$opts -$arg"
35 done

37 shift $(( $OPTIND - 1 ))

39 if [ $# -le 3 ]; then
40     echo "$0: at least 3 arguments required." >&2
41     exit 1
42 fi

44 old="$1"; shift
45 new="$1"; shift
46 differ="$1"; shift

48 errlog=/tmp/protocmp.err.$$

50 protocmp $opts "$@" 2>$errlog |
51 /usr/xpg4/bin/awk -v old="$old" -v new="$new" -v differ="$differ" '
52 nawk -v old="$old" -v new="$new" -v differ="$differ" '
53 /^\*\$/ {
54     next;
55 }
56 /^\* Entries/ {
57     category++;
58     next;
59 }
59 /^\* filea ==/ {
```

new/usr/src/tools/scripts/protocmp.terse.sh

2

```
60     filea = $NF;
61     next;
62 }
63 /^\* fileb ==/ {
64     fileb = $NF;
65     next;
66 }
67 {
68     buf[category, ++line[category]] = $0
69 }
70 END {
71     if (line[1] > 2) {
72         printf("\n%s\n", old);
73         for (i = 1; i <= line[1]; i++) {
74             print buf[1, i];
75         }
76     }
77     if (line[2] > 2) {
78         printf("\n%s\n", new);
79         for (i = 1; i <= line[2]; i++) {
80             print buf[2, i];
81         }
82     }
83     if (line[3] > 2) {
84         printf("\n%s\n", differ);
85         for (i = 1; i <= line[3]; i++) {
86             print buf[3, i];
87         }
88     }
89 }'

91 if [ -s $errlog ]; then
92     echo "\n==== protocmp ERRORS =====\n"
93     cat $errlog
94 fi
95 rm -f $errlog
```

new/usr/src/tools/scripts/webrev.sh

1

```
*****
91242 Sun May 26 18:23:54 2013
new/usr/src/tools/scripts/webrev.sh
XXX Remove awk(1)
*****
_____unchanged_portion_omitted_____

926 #
927 # fix_postscript
928 #

1433 #
1434 # comments_from_teamware {text|html} parent-file child-file
1435 #
1436 # Find the first delta in the child that's not in the parent.  Get the
1437 # newest delta from the parent, get all deltas from the child starting
1438 # with that delta, and then get all info starting with the second oldest
1439 # delta in that list (the first delta unique to the child).
1440 #
1441 # This code adapted from Bill Shannon's "spc" script
1442 #
1443 comments_from_teamware()
1444 {
1445     fmt=$1
1446     pfile=$PWS/$2
1447     cfile=$CWS/$3

1449     if [[ ! -f $PWS/${2%*/}/SCCS/s.${2##*/} ]] && -n $RWS ]; then
1450         pfile=$RWS/$2
1451     fi

1453     if [[ -f $pfile ]]; then
1454         psid=$(SCCS prs -d:I: $pfile 2>/dev/null)
1455     else
1456         psid=1.1
1457     fi

1459     set -A sids $(SCCS prs -l -r$psid -d:I: $cfile 2>/dev/null)
1460     N=${#sids[@]}

1462     awkprg='
1462     awkprg='
1463     /^COMMENTS:/ {p=1; continue}
1464     /^D [0-9]+\.[0-9]+/ {printf "--- %s ---\n", $2; p=0; }
1465     NF == 0u { continue }
1466     {if (p==0) continue; print $0 }'

1468     if [[ $N -ge 2 ]]; then
1469         sidl=${sids[${N-2}]} # Gets 2nd to last sid

1471         if [[ $fmt == "text" ]]; then
1472             SCCS prs -l -r$sidl $cfile 2>/dev/null | \
1473                 $AWK "$awkprg"
1473                 $AWK "$sawkprg"
1474             return
1475         fi

1477         SCCS prs -l -r$sidl $cfile 2>/dev/null | \
1478             html_quote | its2url | $AWK "$sawkprg"
1478             html_quote | its2url | $AWK "$sawkprg"
1479     fi
1480 }
_____unchanged_portion_omitted_____

2259 #
```

new/usr/src/tools/scripts/webrev.sh

2

```
2260 #
2261 # Main program starts here
2262 #
2263 #

2265 trap "rm -f /tmp/$$.* ; exit" 0 1 2 3 15

2267 set +o noclobber

2269 PATH=$(/bin/dirname "$(whence $0)"):$PATH

2271 [[ -z $WDIFF ]] && WDIFF='look_for_prog wdiff'
2272 [[ -z $WX ]] && WX='look_for_prog wx'
2273 [[ -z $HG_ACTIVE ]] && HG_ACTIVE='look_for_prog hg-active'
2274 [[ -z $GIT ]] && GIT='look_for_prog git'
2275 [[ -z $WHICH_SCM ]] && WHICH_SCM='look_for_prog which_scm'
2276 [[ -z $CODEREVIEW ]] && CODEREVIEW='look_for_prog codereview'
2277 [[ -z $PS2PDF ]] && PS2PDF='look_for_prog ps2pdf'
2278 [[ -z $PERL ]] && PERL='look_for_prog perl'
2279 [[ -z $RSYNC ]] && RSYNC='look_for_prog rsync'
2280 [[ -z $SCCS ]] && SCCS='look_for_prog sccs'
2281 [[ -z $AWK ]] && AWK='look_for_prog awk'
2282 [[ -z $GAWK ]] && GAWK='look_for_prog gawk'
2283 [[ -z $AWK ]] && AWK='look_for_prog awk'
2284 [[ -z $SCP ]] && SCP='look_for_prog scp'
2285 [[ -z $SED ]] && SED='look_for_prog sed'
2286 [[ -z $SFTP ]] && SFTP='look_for_prog sftp'
2287 [[ -z $SORT ]] && SORT='look_for_prog sort'
2288 [[ -z $MKTEMP ]] && MKTEMP='look_for_prog mktemp'
2289 [[ -z $GREP ]] && GREP='look_for_prog grep'
2290 [[ -z $FIND ]] && FIND='look_for_prog find'

2292 # set name of trash directory for remote webrev deletion
2293 TRASH_DIR=".trash"
2294 [[ -n $WEBREV_TRASH_DIR ]] && TRASH_DIR=$WEBREV_TRASH_DIR

2296 if [[ ! -x $PERL ]]; then
2297     print -u2 "Error: No perl interpreter found.  Exiting."
2298     exit 1
2299 fi

2301 if [[ ! -x $WHICH_SCM ]]; then
2302     print -u2 "Error: Could not find which_scm.  Exiting."
2303     exit 1
2304 fi

2306 #
2307 # These aren't fatal, but we want to note them to the user.
2308 # We don't warn on the absence of 'wx' until later when we've
2309 # determined that we actually need to try to invoke it.
2310 #
2311 [[ ! -x $CODEREVIEW ]] && print -u2 "WARNING: codereview(1) not found."
2312 [[ ! -x $PS2PDF ]] && print -u2 "WARNING: ps2pdf(1) not found."
2313 [[ ! -x $WDIFF ]] && print -u2 "WARNING: wdiff not found."

2315 # Declare global total counters.
2316 integer TOTL TINS TDEL TMOD TUNC

2318 # default remote host for upload/delete
2319 typeset -r DEFAULT_REMOTE_HOST="cr.opensolaris.org"
2320 # prefixes for upload targets
2321 typeset -r rsync_prefix="rsync://"
2322 typeset -r ssh_prefix="ssh://"

2324 Cflag=
2325 Dflag=
```

new/usr/src/tools/scripts/webrev.sh

3

```

2326 flist_mode=
2327 flist_file=
2328 iflag=
2329 Iflag=
2330 lflag=
2331 Nflag=
2332 nflag=
2333 Oflag=
2334 oflag=
2335 pflag=
2336 tflag=
2337 uflag=
2338 Uflag=
2339 wflag=
2340 remote_target=

2342 #
2343 # NOTE: when adding/removing options it is necessary to sync the list
2344 #       with usr/src/tools/onbld/hgext/cdm.py
2345 #
2346 while getopts "C:Di:I:lnNo:Op:t:Uw" opt
2347 do
2348     case $opt in
2349     C)      Cflag=1
2350            ITSCONF=$OPTARG;;
2352     D)      Dflag=1;;
2354     i)      iflag=1
2355            INCLUDE_FILE=$OPTARG;;
2357     I)      Iflag=1
2358            ITSREG=$OPTARG;;

2360     #
2361     # If -l has been specified, we need to abort further options
2362     # processing, because subsequent arguments are going to be
2363     # arguments to 'putback -n'.
2364     #
2365     l)      lflag=1
2366            break;;

2368     N)      Nflag=1;;

2370     n)      nflag=1;;

2372     O)      Oflag=1;;

2374     o)      oflag=1
2375            # Strip the trailing slash to correctly form remote target.
2376            WDIR=${OPTARG%/*};;

2378     p)      pflag=1
2379            codemgr_parent=$OPTARG;;

2381     t)      tflag=1
2382            remote_target=$OPTARG;;

2384     U)      Uflag=1;;

2386     w)      wflag=1;;

2388     ?)      usage;;
2389     esac
2390 done

```

new/usr/src/tools/scripts/webrev.sh

4

```

2392 FLIST=/tmp/$$flist

2394 if [[ -n $wflag && -n $lflag ]]; then
2395     usage
2396 fi

2398 # more sanity checking
2399 if [[ -n $nflag && -z $Uflag ]]; then
2400     print "it does not make sense to skip webrev generation" \
2401         "without -U"
2402     exit 1
2403 fi

2405 if [[ -n $tflag && -z $Uflag && -z $Dflag ]]; then
2406     echo "remote target has to be used only for upload or delete"
2407     exit 1
2408 fi

2410 #
2411 # For the invocation "webrev -n -U" with no other options, webrev will assume
2412 # that the webrev exists in ${CWS}/webrev, but will upload it using the name
2413 # $(basename ${CWS}). So we need to get CWS set before we skip any remaining
2414 # logic.
2415 #
2416 $WHICH_SCM | read SCM_MODE junk || exit 1
2417 if [[ $SCM_MODE == "teamware" ]]; then
2418     #
2419     # Teamware priorities:
2420     # 1. CODEMGR_WS from the environment
2421     # 2. workspace name
2422     #
2423     [[ -z $codemgr_ws && -n $CODEMGR_WS ]] && codemgr_ws=$CODEMGR_WS
2424     if [[ -n $codemgr_ws && ! -d $codemgr_ws ]]; then
2425         print -u2 "$codemgr_ws: no such workspace"
2426         exit 1
2427     fi
2428     [[ -z $codemgr_ws ]] && codemgr_ws=$(workspace name)
2429     codemgr_ws=$(cd $codemgr_ws;print $PWD)
2430     CODEMGR_WS=$codemgr_ws
2431     CWS=$codemgr_ws
2432 elif [[ $SCM_MODE == "mercurial" ]]; then
2433     #
2434     # Mercurial priorities:
2435     # 1. hg root from CODEMGR_WS environment variable
2436     # 1a. hg root from CODEMGR_WS/usr/closed if we're somewhere under
2437     #     usr/closed when we run webrev
2438     # 2. hg root from directory of invocation
2439     #
2440     if [[ ${PWD} =~ "usr/closed" ]]; then
2441         testparent=${CODEMGR_WS}/usr/closed
2442         # If we're in OpenSolaris mode, we enforce a minor policy:
2443         # help to make sure the reviewer doesn't accidentally publish
2444         # source which is under usr/closed
2445         if [[ -n "$Oflag" ]]; then
2446             print -u2 "OpenSolaris output not permitted with" \
2447                 "usr/closed changes"
2448             exit 1
2449         fi
2450     else
2451         testparent=${CODEMGR_WS}
2452     fi
2453     [[ -z $codemgr_ws && -n $testparent ]] && \
2454         codemgr_ws=$(hg root -R $testparent 2>/dev/null)
2455     [[ -z $codemgr_ws ]] && codemgr_ws=$(hg root 2>/dev/null)
2456     CWS=$codemgr_ws
2457 elif [[ $SCM_MODE == "git" ]]; then

```

```

2458 #
2459 # Git priorities:
2460 # 1. git rev-parse --git-dir from CODEMGR_WS environment variable
2461 # 2. git rev-parse --git-dir from directory of invocation
2462 #
2463 [[ -z $codemgr_ws && -n $CODEMGR_WS ]] && \
2464   codemgr_ws=$(GIT --git-dir=$CODEMGR_WS/.git rev-parse --git-dir \
2465     2>/dev/null)
2466 [[ -z $codemgr_ws ]] && \
2467   codemgr_ws=$(GIT rev-parse --git-dir 2>/dev/null)
2469 if [[ "$codemgr_ws" == ".git" ]]; then
2470     codemgr_ws="{PWD}/${codemgr_ws}"
2471 fi
2473 codemgr_ws=$(dirname $codemgr_ws) # Lose the './.git'
2474 CWS="$codemgr_ws"
2475 elif [[ $SCM_MODE == "subversion" ]]; then
2476 #
2477 # Subversion priorities:
2478 # 1. CODEMGR_WS from environment
2479 # 2. Relative path from current directory to SVN repository root
2480 #
2481 if [[ -n $CODEMGR_WS && -d $CODEMGR_WS/.svn ]]; then
2482     CWS=$CODEMGR_WS
2483 else
2484     svn info | while read line; do
2485         if [[ $line == "URL: *" ]]; then
2486             url=${line#URL: }
2487             elif [[ $line == "Repository Root: *" ]]; then
2488                 repo=${line#Repository Root: }
2489             fi
2490         done
2492         rel=${url#$repo}
2493         CWS="{PWD%$rel}"
2494     fi
2495 fi
2497 #
2498 # If no SCM has been determined, take either the environment setting
2499 # setting for CODEMGR_WS, or the current directory if that wasn't set.
2500 #
2501 if [[ -z $CWS ]]; then
2502     CWS=${CODEMGR_WS:-.}
2503 fi
2505 #
2506 # If the command line options indicate no webrev generation, either
2507 # explicitly (-n) or implicitly (-D but not -U), then there's a whole
2508 # ton of logic we can skip.
2509 #
2510 # Instead of increasing indentation, we intentionally leave this loop
2511 # body open here, and exit via break from multiple points within.
2512 # Search for DO_EVERYTHING below to find the break points and closure.
2513 #
2514 for do_everything in 1; do
2516 # DO_EVERYTHING: break point
2517 if [[ -n $nflag || ( -z $Uflag && -n $Dflag ) ]]; then
2518     break
2519 fi
2521 #
2522 # If this manually set as the parent, and it appears to be an earlier webrev,
2523 # then note that fact and set the parent to the raw_files/new subdirectory.

```

```

2524 #
2525 if [[ -n $pflag && -d $codemgr_parent/raw_files/new ]]; then
2526     parent_webrev=$(readlink -f "$codemgr_parent")
2527     codemgr_parent=$(readlink -f "$codemgr_parent/raw_files/new")
2528 fi
2530 if [[ -z $wflag && -z $lflag ]]; then
2531     shift $((SOPTIND - 1))
2533     if [[ $1 == "-" ]]; then
2534         cat > $FLIST
2535         flist_mode="stdin"
2536         flist_done=1
2537         shift
2538     elif [[ -n $1 ]]; then
2539         if [[ ! -r $1 ]]; then
2540             print -u2 "$1: no such file or not readable"
2541             usage
2542         fi
2543         cat $1 > $FLIST
2544         flist_mode="file"
2545         flist_file=$1
2546         flist_done=1
2547         shift
2548     else
2549         flist_mode="auto"
2550     fi
2551 fi
2553 #
2554 # Before we go on to further consider -l and -w, work out which SCM we think
2555 # is in use.
2556 #
2557 case "$SCM_MODE" in
2558     teamware|mercurial|git|subversion)
2559         ;;
2560     unknown)
2561         if [[ $flist_mode == "auto" ]]; then
2562             print -u2 "Unable to determine SCM in use and file list not spec
2563             print -u2 "See which_scm(1) for SCM detection information."
2564             exit 1
2565         fi
2566         ;;
2567     *)
2568         if [[ $flist_mode == "auto" ]]; then
2569             print -u2 "Unsupported SCM in use ($SCM_MODE) and file list not
2570             exit 1
2571         fi
2572         ;;
2573 esac
2575 print -u2 "    SCM detected: $SCM_MODE"
2577 if [[ -n $lflag ]]; then
2578 #
2579 # If the -l flag is given instead of the name of a file list,
2580 # then generate the file list by extracting file names from a
2581 # putback -n.
2582 #
2583 shift $((SOPTIND - 1))
2584 if [[ $SCM_MODE == "teamware" ]]; then
2585     flist_from_teamware "$*"
2586 else
2587     print -u2 -- "Error: -l option only applies to TeamWare"
2588     exit 1
2589 fi

```

```

2590     flist_done=1
2591     shift $#
2592 elif [[ -n $wflag ]]; then
2593     #
2594     # If the -w is given then assume the file list is in Bonwick's "wx"
2595     # command format, i.e. pathname lines alternating with SCCS comment
2596     # lines with blank lines as separators. Use the SCCS comments later
2597     # in building the index.html file.
2598     #
2599     shift $(( $OPTIND - 1 ))
2600     wxfile=$1
2601     if [[ -z $wxfile && -n $CODEMGR_WS ]]; then
2602         if [[ -r $CODEMGR_WS/wx/active ]]; then
2603             wxfile=$CODEMGR_WS/wx/active
2604         fi
2605     fi

2607     [[ -z $wxfile ]] && print -u2 "wx file not specified, and could not " \
2608     "be auto-detected (check \$CODEMGR_WS)" && exit 1

2610     if [[ ! -r $wxfile ]]; then
2611         print -u2 "$wxfile: no such file or not readable"
2612         usage
2613     fi

2615     print -u2 " File list from: wx 'active' file '$wxfile' ... \c"
2616     flist_from_wx $wxfile
2617     flist_done=1
2618     if [[ -n "$*" ]]; then
2619         shift
2620     fi
2621 elif [[ $flist_mode == "stdin" ]]; then
2622     print -u2 " File list from: standard input"
2623 elif [[ $flist_mode == "file" ]]; then
2624     print -u2 " File list from: $flist_file"
2625 fi

2627 if [[ $# -gt 0 ]]; then
2628     print -u2 "WARNING: unused arguments: $"
2629 fi

2631 #
2632 # Before we entered the DO EVERYTHING loop, we should have already set CWS
2633 # and CODEMGR_WS as needed. Here, we set the parent workspace.
2634 #

2636 if [[ $SCM_MODE == "teamware" ]]; then

2638     #
2639     # Teamware priorities:
2640     #
2641     # 1) via -p command line option
2642     # 2) in the user environment
2643     # 3) in the flist
2644     # 4) automatically based on the workspace
2645     #

2647     #
2648     # For 1, codemgr_parent will already be set. Here's 2:
2649     #
2650     [[ -z $codemgr_parent && -n $CODEMGR_PARENT ]] && \
2651     codemgr_parent=$CODEMGR_PARENT
2652     if [[ -n $codemgr_parent && ! -d $codemgr_parent ]]; then
2653         print -u2 "$codemgr_parent: no such directory"
2654         exit 1
2655     fi

```

```

2657     #
2658     # If we're in auto-detect mode and we haven't already gotten the file
2659     # list, then see if we can get it by probing for wx.
2660     #
2661     if [[ -z $flist_done && $flist_mode == "auto" && -n $codemgr_ws ]]; then
2662         if [[ ! -x $WX ]]; then
2663             print -u2 "WARNING: wx not found!"
2664         fi

2666     #
2667     # We need to use wx list -w so that we get renamed files, etc.
2668     # but only if a wx active file exists-- otherwise wx will
2669     # hang asking us to initialize our wx information.
2670     #
2671     if [[ -x $WX && -f $codemgr_ws/wx/active ]]; then
2672         print -u2 " File list from: 'wx list -w' ... \c"
2673         $WX list -w > $FLIST
2674         $WX comments > /tmp/$$.wx_comments
2675         wxfile=/tmp/$$.wx_comments
2676         print -u2 "done"
2677         flist_done=1
2678     fi
2679 fi

2681 #
2682 # If by hook or by crook we've gotten a file list by now (perhaps
2683 # from the command line), eval it to extract environment variables from
2684 # it: This is method 3 for finding the parent.
2685 #
2686 if [[ -z $flist_done ]]; then
2687     flist_from_teamware
2688 fi
2689 env_from_flist

2691 #
2692 # (4) If we still don't have a value for codemgr_parent, get it
2693 # from workspace.
2694 #
2695 [[ -z $codemgr_parent ]] && codemgr_parent='workspace parent'
2696 if [[ ! -d $codemgr_parent ]]; then
2697     print -u2 "$CODEMGR_PARENT: no such parent workspace"
2698     exit 1
2699 fi

2701 PWS=$codemgr_parent

2703 [[ -n $parent_webrev ]] && RWS=$(workspace parent $CWS)

2705 elif [[ $SCM_MODE == "mercurial" ]]; then
2706     #
2707     # Parent can either be specified with -p
2708     # Specified with CODEMGR_PARENT in the environment
2709     # or taken from hg's default path.
2710     #

2712     if [[ -z $codemgr_parent && -n $CODEMGR_PARENT ]]; then
2713         codemgr_parent=$CODEMGR_PARENT
2714     fi

2716     if [[ -z $codemgr_parent ]]; then
2717         codemgr_parent='hg path -R $codemgr_ws default 2>/dev/null'
2718     fi

2720     PWS=$codemgr_parent

```

```

2722 #
2723 # If the parent is a webrev, we want to do some things against
2724 # the natural workspace parent (file list, comments, etc)
2725 #
2726 if [[ -n $parent_webrev ]]; then
2727     real_parent=$(hg path -R $codemgr_ws default 2>/dev/null)
2728 else
2729     real_parent=$PWS
2730 fi
2731
2732 #
2733 # If hg-active exists, then we run it. In the case of no explicit
2734 # flist given, we'll use it for our comments. In the case of an
2735 # explicit flist given we'll try to use it for comments for any
2736 # files mentioned in the flist.
2737 #
2738 if [[ -z $flist_done ]]; then
2739     flist_from_mercurial $CWS $real_parent
2740     flist_done=1
2741 fi
2742
2743 #
2744 # If we have a file list now, pull out any variables set
2745 # therein. We do this now (rather than when we possibly use
2746 # hg-active to find comments) to avoid stomping specifications
2747 # in the user-specified flist.
2748 #
2749 if [[ -n $flist_done ]]; then
2750     env_from_flist
2751 fi
2752
2753 #
2754 # Only call hg-active if we don't have a wx formatted file already
2755 #
2756 if [[ -x $HG_ACTIVE && -z $wxfile ]]; then
2757     print " Comments from: hg-active -p $real_parent ...\c"
2758     hg_active_wxfile $CWS $real_parent
2759     print " Done."
2760 fi
2761
2762 #
2763 # At this point we must have a wx flist either from hg-active,
2764 # or in general. Use it to try and find our parent revision,
2765 # if we don't have one.
2766 #
2767 if [[ -z $HG_PARENT ]]; then
2768     eval `sSED -e "s/#.*$//" $wxfile | $GREP HG_PARENT=`
2769 fi
2770
2771 #
2772 # If we still don't have a parent, we must have been given a
2773 # wx-style active list with no HG_PARENT specification, run
2774 # hg-active and pull an HG_PARENT out of it, ignore the rest.
2775 #
2776 if [[ -z $HG_PARENT && -x $HG_ACTIVE ]]; then
2777     SHG_ACTIVE -w $codemgr_ws -p $real_parent | \
2778     eval `sSED -e "s/#.*$//" | $GREP HG_PARENT=`
2779 elif [[ -z $HG_PARENT ]]; then
2780     print -u2 "Error: Cannot discover parent revision"
2781     exit 1
2782 fi
2783
2784 pnode=$(trim_digest $HG_PARENT)
2785 PRETTY_PWS=${PWS} (at ${pnode})"
2786 cnode=$(hg parent -R $codemgr_ws --template '{node|short}' \
2787     2>/dev/null)

```

```

2788     PRETTY_CWS=${CWS} (at ${cnode}))"
2789 elif [[ $SCM_MODE == "git" ]]; then
2790     #
2791     # Parent can either be specified with -p, or specified with
2792     # CODEMGR_PARENT in the environment.
2793     #
2794
2795     if [[ -z $codemgr_parent && -n $CODEMGR_PARENT ]]; then
2796         codemgr_parent=$CODEMGR_PARENT
2797     fi
2798
2799     # Try to figure out the parent based on the branch the current
2800     # branch is tracking, if we fail, use origin/master
2801     this_branch=$(GIT branch | /usr/xpg4/bin/awk '$1 == "*" { print $2 }')
2802     this_branch=$(GIT branch | awk '$1 == "*" { print $2 }')
2803     par_branch="origin/master"
2804
2805     # If we're not on a branch there's nothing we can do
2806     if [[ $this_branch != "(no branch)" ]]; then
2807         $GIT for-each-ref
2808         --format='%(refname:short) %(upstream:short)' refs/heads/ |
2809         while read local remote; do
2810             [[ "$local" == "$this_branch" ]] && par_branch="$remote"
2811         done
2812     fi
2813
2814     if [[ -z $codemgr_parent ]]; then
2815         codemgr_parent=$par_branch
2816     fi
2817     PWS=$codemgr_parent
2818
2819 #
2820 # If the parent is a webrev, we want to do some things against
2821 # the natural workspace parent (file list, comments, etc)
2822 #
2823 if [[ -n $parent_webrev ]]; then
2824     real_parent=$par_branch
2825 else
2826     real_parent=$PWS
2827 fi
2828
2829 if [[ -z $flist_done ]]; then
2830     flist_from_git "$CWS" "$real_parent"
2831     flist_done=1
2832 fi
2833
2834 #
2835 # If we have a file list now, pull out any variables set
2836 # therein.
2837 #
2838 if [[ -n $flist_done ]]; then
2839     env_from_flist
2840 fi
2841
2842 #
2843 # If we don't have a wx-format file list, build one we can pull change
2844 # comments from.
2845 #
2846 if [[ -z $wxfile ]]; then
2847     print " Comments from: git...\c"
2848     git_wxfile "$CWS" "$real_parent"
2849     print " Done."
2850 fi
2851
2852 if [[ -z $GIT_PARENT ]]; then
2853     GIT_PARENT=$(GIT merge-base "$real_parent" HEAD)

```

```

2853     fi
2854     if [[ -z $GIT_PARENT ]]; then
2855         print -u2 "Error: Cannot discover parent revision"
2856         exit 1
2857     fi

2859     pnode=$(trim_digest $GIT_PARENT)

2861     if [[ $real_parent == */* ]]; then
2862         origin=$(echo $real_parent | cut -d/ -f1)
2863         origin=${GIT_remote -v | \
2864             $AWK ' $1 == "$origin" { print $2; exit }')
2865         PRETTY_PWS="{PWS} ({origin} at {pnode})"
2866     else
2867         PRETTY_PWS="{PWS} (at {pnode})"
2868     fi

2870     cnode=${GIT --git-dir=${codemgr_ws}/.git rev-parse --short=12 HEAD \
2871         2>/dev/null)
2872     PRETTY_CWS="{CWS} (at {cnode})"
2873     elif [[ $SCM_MODE == "subversion" ]]; then

2875     #
2876     # We only will have a real parent workspace in the case one
2877     # was specified (be it an older webrev, or another checkout).
2878     #
2879     [[ -n $codemgr_parent ]] && PWS=$codemgr_parent

2881     if [[ -z $flist_done && $flist_mode == "auto" ]]; then
2882         flist_from_subversion $CWS $OLDPWD
2883     fi
2884 else
2885     if [[ $SCM_MODE == "unknown" ]]; then
2886         print -u2 "    Unknown type of SCM in use"
2887     else
2888         print -u2 "    Unsupported SCM in use: $SCM_MODE"
2889     fi

2891     env_from_flist

2893     if [[ -z $CODEMGR_WS ]]; then
2894         print -u2 "SCM not detected/supported and CODEMGR_WS not specified"
2895         exit 1
2896     fi

2898     if [[ -z $CODEMGR_PARENT ]]; then
2899         print -u2 "SCM not detected/supported and CODEMGR_PARENT not specified"
2900         exit 1
2901     fi

2903     CWS=$CODEMGR_WS
2904     PWS=$CODEMGR_PARENT
2905 fi

2907 #
2908 # If the user didn't specify a -i option, check to see if there is a
2909 # webrev-info file in the workspace directory.
2910 #
2911 if [[ -z $iflag && -r "$CWS/webrev-info" ]]; then
2912     iflag=1
2913     INCLUDE_FILE="$CWS/webrev-info"
2914 fi

2916 if [[ -n $iflag ]]; then
2917     if [[ ! -r $INCLUDE_FILE ]]; then
2918         print -u2 "include file '$INCLUDE_FILE' does not exist or is" \

```

```

2919         "not readable."
2920         exit 1
2921     else
2922         #
2923         # $INCLUDE_FILE may be a relative path, and the script alters
2924         # PWD, so we just stash a copy in /tmp.
2925         #
2926         cp $INCLUDE_FILE /tmp/${$.include}
2927     fi
2928 fi

2930 # DO EVERYTHING: break point
2931 if [[ -n $Nflag ]]; then
2932     break
2933 fi

2935 typeset -A itsinfo
2936 typeset -r its_sed_script=/tmp/${$.its_sed}
2937 valid_prefixes=
2938 if [[ -z $nflag ]]; then
2939     DEFREGFILE="$(/bin/dirname "$(whence $0)"/../etc/its.reg)"
2940     if [[ -n $iflag ]]; then
2941         REGFILE=$ITSREG
2942     elif [[ -r $HOME/.its.reg ]]; then
2943         REGFILE=$HOME/.its.reg
2944     else
2945         REGFILE=$DEFREGFILE
2946     fi
2947     if [[ ! -r $REGFILE ]]; then
2948         print "ERROR: Unable to read database registry file $REGFILE"
2949         exit 1
2950     elif [[ $REGFILE != $DEFREGFILE ]]; then
2951         print "    its.reg from: $REGFILE"
2952     fi

2954     $$SED -e '/^#/d' -e '/^[          ]*$/d' $REGFILE | while read LINE; do

2956         name=${LINE%%=*}
2957         value=${LINE#*=}

2959         if [[ $name == PREFIX ]]; then
2960             p=${value}
2961             valid_prefixes="{p} ${valid_prefixes}"
2962         else
2963             itsinfo["${p}_${name}"]="${value}"
2964         fi
2965     done

2968     DEFCONFFILE="$(/bin/dirname "$(whence $0)"/../etc/its.conf)"
2969     CONFFILES=$DEFCONFFILE
2970     if [[ -r $HOME/.its.conf ]]; then
2971         CONFFILES="{CONFFILES} $HOME/.its.conf"
2972     fi
2973     if [[ -n $Cflag ]]; then
2974         CONFFILES="{CONFFILES} ${ITSCONF}"
2975     fi
2976     its_domain=
2977     its_priority=
2978     for cf in ${CONFFILES}; do
2979         if [[ ! -r $cf ]]; then
2980             print "ERROR: Unable to read database configuration file"
2981             exit 1
2982         elif [[ $cf != $DEFCONFFILE ]]; then
2983             print "    its.conf: reading $cf"
2984         fi

```

```

2985         SSED -e '/^#/d' -e '/^[      ]*/d' $cf | while read LINE; do
2986             eval "${LINE}"
2987         done
2988     done

2990 #
2991 # If an information tracking system is explicitly identified by prefix,
2992 # we want to disregard the specified priorities and resolve it according
2993 #
2994 # To that end, we'll build a sed script to do each valid prefix in turn.
2995 #
2996 for p in ${valid_prefixes}; do
2997     #
2998     # When an informational URL was provided, translate it to a
2999     # hyperlink.  When omitted, simply use the prefix text.
3000     #
3001     if [[ -z ${itsinfo["${p}_INFO"]} ]]; then
3002         itsinfo["${p}_INFO"]=${p}
3003     else
3004         itsinfo["${p}_INFO"]="

```

```

3051 # Search for DO_EVERYTHING above for matching "for" statement
3052 # and explanation of this terminator.
3053 #
3054 done

3056 #
3057 # Output directory.
3058 #
3059 WDIR=${WDIR:-$CWS/webrev}

3061 #
3062 # Name of the webrev, derived from the workspace name or output directory;
3063 # in the future this could potentially be an option.
3064 #
3065 if [[ -n $oflag ]]; then
3066     WNAME=${WDIR##*/}
3067 else
3068     WNAME=${CWS##*/}
3069 fi

3071 # Make sure remote target is well formed for remote upload/delete.
3072 if [[ -n $Dflag || -n $Uflag ]]; then
3073     #
3074     # If remote target is not specified, build it from scratch using
3075     # the default values.
3076     #
3077     if [[ -z $tflag ]]; then
3078         remote_target=${DEFAULT_REMOTE_HOST}:${WNAME}
3079     else
3080         #
3081         # Check upload target prefix first.
3082         #
3083         if [[ "${remote_target}" != ${rsync_prefix}* &&
3084             "${remote_target}" != ${ssh_prefix}* ]]; then
3085             print "ERROR: invalid prefix of upload URI" \
3086                 "(${remote_target})"
3087             exit 1
3088         fi
3089         #
3090         # If destination specification is not in the form of
3091         # host_spec:remote_dir then assume it is just remote hostname
3092         # and append a colon and destination directory formed from
3093         # local webrev directory name.
3094         #
3095         typeset target_no_prefix=${remote_target##*/}
3096         if [[ ${target_no_prefix} == /* ]]; then
3097             if [[ "${remote_target}" == /* ]]; then
3098                 remote_target=${remote_target}${WNAME}
3099             fi
3100         else
3101             if [[ ${target_no_prefix} == /* ]]; then
3102                 print "ERROR: badly formed upload URI" \
3103                     "(${remote_target})"
3104                 exit 1
3105             else
3106                 remote_target=${remote_target}:${WNAME}
3107             fi
3108         fi
3109     fi

3111 #
3112 # Strip trailing slash.  Each upload method will deal with directory
3113 # specification separately.
3114 #
3115 remote_target=${remote_target%/}
3116 fi

```



```

3118 #
3119 # Option -D by itself (option -U not present) implies no webrev generation.
3120 #
3121 if [[ -z $Uflag && -n $Dflag ]]; then
3122     delete_webrev 1 1
3123     exit $?
3124 fi

3126 #
3127 # Do not generate the webrev, just upload it or delete it.
3128 #
3129 if [[ -n $nflag ]]; then
3130     if [[ -n $Dflag ]]; then
3131         delete_webrev 1 1
3132         (( $? == 0 )) || exit $?
3133     fi
3134     if [[ -n $Uflag ]]; then
3135         upload_webrev
3136         exit $?
3137     fi
3138 fi

3140 if [ "${WDIR%/*}" ]; then
3141     WDIR=$PWD/$WDIR
3142 fi

3144 if [[ ! -d $WDIR ]]; then
3145     mkdir -p $WDIR
3146     (( $? != 0 )) && exit 1
3147 fi

3149 #
3150 # Summarize what we're going to do.
3151 #
3152 print "    Workspace: ${PRETTY_CWS:-$CWS}"
3153 if [[ -n $parent_webrev ]]; then
3154     print "Compare against: webrev at $parent_webrev"
3155 else
3156     print "Compare against: ${PRETTY_PWS:-$PWS}"
3157 fi

3159 [[ -n $INCLUDE_FILE ]] && print "    Including: $INCLUDE_FILE"
3160 print "    Output to: $WDIR"

3162 #
3163 # Save the file list in the webrev dir
3164 #
3165 [[ ! $FLIST -ef $WDIR/file.list ]] && cp $FLIST $WDIR/file.list

3167 rm -f $WDIR/$WNAME.patch
3168 rm -f $WDIR/$WNAME.ps
3169 rm -f $WDIR/$WNAME.pdf

3171 touch $WDIR/$WNAME.patch

3173 print "    Output Files:"

3175 #
3176 # Clean up the file list: Remove comments, blank lines and env variables.
3177 #
3178 $SED -e "s/#.*$//" -e "/=/d" -e "/^[ ]*$/" $FLIST > /tmp/$$.flist.clean
3179 FLIST=/tmp/$$.flist.clean

3181 #
3182 # For Mercurial, create a cache of manifest entries.

```

```

3183 #
3184 if [[ $SCM_MODE == "mercurial" ]]; then
3185     #
3186     # Transform the FLIST into a temporary sed script that matches
3187     # relevant entries in the Mercurial manifest as follows:
3188     # 1) The script will be used against the parent revision manifest,
3189     #    so for FLIST lines that have two filenames (a renamed file)
3190     #    keep only the old name.
3191     # 2) Escape all forward slashes the filename.
3192     # 3) Change the filename into another sed command that matches
3193     #    that file in "hg manifest -v" output: start of line, three
3194     #    octal digits for file permissions, space, a file type flag
3195     #    character, space, the filename, end of line.
3196     # 4) Eliminate any duplicate entries. (This can occur if a
3197     #    file has been used as the source of an hg cp and it's
3198     #    also been modified in the same changeset.)
3199     #
3200     SEDFILE=/tmp/$$.manifest.sed
3201     $SED '
3202         s#[^ ]* ##
3203         s/#\\\/#g
3204         s#^.*$#/^... . &$/p#
3205     ' < $FLIST | $SORT -u > $SEDFILE

3207     #
3208     # Apply the generated script to the output of "hg manifest -v"
3209     # to get the relevant subset for this webrev.
3210     #
3211     HG_PARENT_MANIFEST=/tmp/$$.manifest
3212     hg -R $CWS manifest -v -r $HG_PARENT |
3213     $SED -n -f $SEDFILE > $HG_PARENT_MANIFEST
3214 fi

3216 #
3217 # First pass through the files: generate the per-file webrev HTML-files.
3218 #
3219 cat $FLIST | while read LINE
3220 do
3221     set - $LINE
3222     P=$1

3224     #
3225     # Normally, each line in the file list is just a pathname of a
3226     # file that has been modified or created in the child. A file
3227     # that is renamed in the child workspace has two names on the
3228     # line: new name followed by the old name.
3229     #
3230     oldname=""
3231     oldpath=""
3232     rename=
3233     if [[ $# -eq 2 ]]; then
3234         PP=$2 # old filename
3235         if [[ -f $PP ]]; then
3236             oldname=" (copied from $PP)"
3237         else
3238             oldname=" (renamed from $PP)"
3239         fi
3240         oldpath="$PP"
3241         rename=1
3242         PDIR=${PP%/*}
3243         if [[ $PDIR == $PP ]]; then
3244             PDIR="." # File at root of workspace
3245         fi

3247         PF=${PP##*/}

```

```

3249     DIR=${P%/*}
3250     if [[ $DIR == $P ]]; then
3251         DIR="." # File at root of workspace
3252     fi
3254     F=${P##*/}
3256 else
3257     DIR=${P%/*}
3258     if [[ "$DIR" == "$P" ]]; then
3259         DIR="." # File at root of workspace
3260     fi
3262     F=${P##*/}
3264     PP=$P
3265     PDIR=$DIR
3266     PF=$F
3267 fi
3269 COMM='getcomments html $P $PP'
3271 print "\t$P$oldname\n\t\tc"
3273 # Make the webrev mirror directory if necessary
3274 mkdir -p $WDIR/$DIR
3276 #
3277 # We stash old and new files into parallel directories in $WDIR
3278 # and do our diffs there. This makes it possible to generate
3279 # clean looking diffs which don't have absolute paths present.
3280 #
3282 build_old_new "$WDIR" "$PWS" "$PDIR" "$PF" "$CWS" "$DIR" "$F" || \
3283     continue
3285 #
3286 # Keep the old PWD around, so we can safely switch back after
3287 # diff generation, such that build_old_new runs in a
3288 # consistent environment.
3289 #
3290 OWD=$PWD
3291 cd $WDIR/raw_files
3292 ofile=old/$PDIR/$PF
3293 nfile=new/$DIR/$F
3295 mv_but_nodiff=
3296 cmp $ofile $nfile > /dev/null 2>&1
3297 if [[ $? == 0 && $rename == 1 ]]; then
3298     mv_but_nodiff=1
3299 fi
3301 #
3302 # If we have old and new versions of the file then run the appropriate
3303 # diffs. This is complicated by a couple of factors:
3304 #
3305 # - renames must be handled specially: we emit a 'remove'
3306 #   diff and an 'add' diff
3307 # - new files and deleted files must be handled specially
3308 # - Solaris patch(lm) can't cope with file creation
3309 #   (and hence renames) as of this writing.
3310 # - To make matters worse, gnu patch doesn't interpret the
3311 #   output of Solaris diff properly when it comes to
3312 #   adds and deletes. We need to do some "cleansing"
3313 #   transformations:
3314 #       [to add a file] @@ -1,0 +X,Y @@ --> @@ -0,0 +X,Y @@

```

```

3315     #       [to del a file] @@ -X,Y +1,0 @@ --> @@ -X,Y +0,0 @@
3316     #
3317     cleanse_rmfile="$SED 's/^\(@@ [0-9+,-]*\) [0-9+,-]* @@$/\1 +0,0 @@/'"
3318     cleanse_newfile="$SED 's/^\(@@ [0-9+,-]*\) \([0-9+,-]* @@\)$/@@ -0,0 \1/'"
3320 rm -f $WDIR/$DIR/$F.patch
3321 if [[ -z $rename ]]; then
3322     if [ ! -f "$ofile" ]; then
3323         diff -u /dev/null $nfile | sh -c "$cleanse_newfile" \
3324             > $WDIR/$DIR/$F.patch
3325     elif [ ! -f "$nfile" ]; then
3326         diff -u $ofile /dev/null | sh -c "$cleanse_rmfile" \
3327             > $WDIR/$DIR/$F.patch
3328     else
3329         diff -u $ofile $nfile > $WDIR/$DIR/$F.patch
3330     fi
3331 else
3332     diff -u $ofile /dev/null | sh -c "$cleanse_rmfile" \
3333         > $WDIR/$DIR/$F.patch
3335     diff -u /dev/null $nfile | sh -c "$cleanse_newfile" \
3336         >> $WDIR/$DIR/$F.patch
3337 fi
3339 #
3340 # Tack the patch we just made onto the accumulated patch for the
3341 # whole wad.
3342 #
3343 cat $WDIR/$DIR/$F.patch >> $WDIR/$WNAME.patch
3345 print " patch\c"
3347 if [[ -f $ofile && -f $nfile && -z $mv_but_nodiff ]]; then
3349     ${CDIFFCMD:-diff -bt -C 5} $ofile $nfile > $WDIR/$DIR/$F.cdifff
3350     diff_to_html $F $DIR/$F "C" "$COMM" < $WDIR/$DIR/$F.cdifff \
3351         > $WDIR/$DIR/$F.cdifff.html
3352     print " cdifff\c"
3354     ${UDIFFCMD:-diff -bt -U 5} $ofile $nfile > $WDIR/$DIR/$F.udifff
3355     diff_to_html $F $DIR/$F "U" "$COMM" < $WDIR/$DIR/$F.udifff \
3356         > $WDIR/$DIR/$F.udifff.html
3358     print " udifff\c"
3360 if [[ -x $WDIFF ]]; then
3361     $WDIFF -c "$COMM" \
3362         -t "$WNAME wdifff $DIR/$F" $ofile $nfile > \
3363         $WDIR/$DIR/$F.wdifff.html 2>/dev/null
3364     if [[ $? -eq 0 ]]; then
3365         print " wdifff\c"
3366     else
3367         print " wdifff[fail]\c"
3368     fi
3369 fi
3371 sdifff_to_html $ofile $nfile $F $DIR "$COMM" \
3372     > $WDIR/$DIR/$F.sdifff.html
3373 print " sdifff\c"
3375 print " frames\c"
3377 rm -f $WDIR/$DIR/$F.cdifff $WDIR/$DIR/$F.udifff
3379 difflines $ofile $nfile > $WDIR/$DIR/$F.count

```

```

3381     elif [[ -f $ofile && -f $nfile && -n $mw_but_nodiff ]]; then
3382         # renamed file: may also have differences
3383         difflines $ofile $nfile > $WDIR/$DIR/$F.count
3384     elif [[ -f $nfile ]]; then
3385         # new file: count added lines
3386         difflines /dev/null $nfile > $WDIR/$DIR/$F.count
3387     elif [[ -f $ofile ]]; then
3388         # old file: count deleted lines
3389         difflines $ofile /dev/null > $WDIR/$DIR/$F.count
3390     fi

3392     #
3393     # Now we generate the postscript for this file. We generate diffs
3394     # only in the event that there is delta, or the file is new (it seems
3395     # tree-killing to print out the contents of deleted files).
3396     #
3397     if [[ -f $nfile ]]; then
3398         ocr=$ofile
3399         [[ ! -f $ofile ]] && ocr=/dev/null

3401         if [[ -z $mw_but_nodiff ]]; then
3402             textcomm='getcomments text $P $PP'
3403             if [[ -x $CODEREVIEW ]]; then
3404                 $CODEREVIEW -y "$textcomm" \
3405                     -e $ocr $nfile \
3406                     > /tmp/$.psfile 2>/dev/null &&
3407                 cat /tmp/$.psfile >> $WDIR/$WNAME.ps
3408             if [[ $? -eq 0 ]]; then
3409                 print " ps\\c"
3410             else
3411                 print " ps[fail]\\c"
3412             fi
3413         fi
3414     fi
3415 fi

3417 if [[ -f $ofile ]]; then
3418     source_to_html Old $PP < $ofile > $WDIR/$DIR/$F-.html
3419     print " old\\c"
3420 fi

3422 if [[ -f $nfile ]]; then
3423     source_to_html New $P < $nfile > $WDIR/$DIR/$F.html
3424     print " new\\c"
3425 fi

3427 cd $OWD
3429 print
3430 done

3432 frame_nav_js > $WDIR/ancnav.js
3433 frame_navigation > $WDIR/ancnav.html

3435 if [[ ! -f $WDIR/$WNAME.ps ]]; then
3436     print " Generating PDF: Skipped: no output available"
3437 elif [[ -x $CODEREVIEW && -x $PS2PDF ]]; then
3438     print " Generating PDF: \\c"
3439     fix_postscript $WDIR/$WNAME.ps | $PS2PDF - > $WDIR/$WNAME.pdf
3440     print " Done."
3441 else
3442     print " Generating PDF: Skipped: missing 'ps2pdf' or 'codereview'"
3443 fi

3445 # If we're in OpenSolaris mode and there's a closed dir under $WDIR,
3446 # delete it - prevent accidental publishing of closed source

```

```

3448 if [[ -n "$Oflag" ]]; then
3449     $FIND $WDIR -type d -name closed -exec /bin/rm -rf {} \;
3450 fi

3452 # Now build the index.html file that contains
3453 # links to the source files and their diffs.

3455 cd $CWS

3457 # Save total changed lines for Code Inspection.
3458 print "$TOTL" > $WDIR/TotalChangedLines

3460 print "         index.html: \\c"
3461 INDEXFILE=$WDIR/index.html
3462 exec 3<&1 # duplicate stdout to FD3.
3463 exec 1<&- # Close stdout.
3464 exec > $INDEXFILE # Open stdout to index file.

3466 print "$HTML<head>$STDHEAD"
3467 print "<title>$WNAME</title>"
3468 print "</head>"
3469 print "<body id=\\\"SUNWwebrev\\\">"
3470 print "<div class=\\\"summary\\\">"
3471 print "<h2>Code Review for $WNAME</h2>"

3473 print "<table>"

3475 #
3476 # Get the preparer's name:
3477 #
3478 # If the SCM detected is Mercurial, and the configuration property
3479 # ui.username is available, use that, but be careful to properly escape
3480 # angle brackets (HTML syntax characters) in the email address.
3481 #
3482 # Otherwise, use the current userid in the form "John Doe (jdoe)", but
3483 # to maintain compatibility with passwd(4), we must support '&' substitutions.
3484 #
3485 preparer=
3486 if [[ "$SCM_MODE" == mercurial ]]; then
3487     preparer='hg showconfig ui.username 2>/dev/null'
3488     if [[ -n "$preparer" ]]; then
3489         preparer="$(echo "$preparer" | html_quote)"
3490     fi
3491 fi
3492 if [[ -z "$preparer" ]]; then
3493     preparer=$(
3494         $PERL -e '
3495             ($login, $pw, $uid, $gid, $quota, $cmt, $gcos) = getpwuid($<);
3496             if ($login) {
3497                 $gcos =~ s/\\&/ucfirst($login)/e;
3498                 printf "%s (%s)\\n", $gcos, $login;
3499             } else {
3500                 printf "(unknown)\\n";
3501             }
3502         '
3503     )
3504 fi

3505 PREPDATE=$(LC_ALL=C /usr/bin/date +%Y-%b-%d %R %z %Z)
3506 print "<tr><th>Prepared by:</th><td>$preparer on $PREPDATE</td></tr>"
3507 print "<tr><th>Workspace:</th><td>${PRETTY_CWS:-$CWS}"
3508 print "</td></tr>"
3509 print "<tr><th>Compare against:</th><td>"
3510 if [[ -n $parent_webrev ]]; then
3511     print "webrev at $parent_webrev"
3512 else

```

```

3513     print "${PRETTY_PWS:-$PWS}"
3514 fi
3515 print "</td></tr>"
3516 print "<tr><th>Summary of changes:</th><td>"
3517 printCI $TOTL $TINS $TDEL $TMOD $TUNC
3518 print "</td></tr>"

3520 if [[ -f $WDIR/$WNAME.patch ]]; then
3521     wpatch_url="$(print $WNAME.patch | url_encode)"
3522     print "<tr><th>Patch of changes:</th><td>"
3523     print "<a href=\"\$wpatch_url\">$WNAME.patch</a></td></tr>"
3524 fi
3525 if [[ -f $WDIR/$WNAME.pdf ]]; then
3526     wpdf_url="$(print $WNAME.pdf | url_encode)"
3527     print "<tr><th>Printable review:</th><td>"
3528     print "<a href=\"\$wpdf_url\">$WNAME.pdf</a></td></tr>"
3529 fi

3531 if [[ -n "$iflag" ]]; then
3532     print "<tr><th>Author comments:</th><td><div>"
3533     cat /tmp/$.include
3534     print "</div></td></tr>"
3535 fi
3536 print "</table>"
3537 print "</div>"

3539 #
3540 # Second pass through the files: generate the rest of the index file
3541 #
3542 cat $FLIST | while read LINE
3543 do
3544     set - $LINE
3545     P=$1

3547     if [[ $# == 2 ]]; then
3548         PP=$2
3549         oldname="$PP"
3550     else
3551         PP=$P
3552         oldname=""
3553     fi

3555     mv_but_nodiff=
3556     cmp $WDIR/raw_files/old/$PP $WDIR/raw_files/new/$P > /dev/null 2>&1
3557     if [[ $? == 0 && -n "$oldname" ]]; then
3558         mv_but_nodiff=1
3559     fi

3561     DIR=${P%/*}
3562     if [[ $DIR == $P ]]; then
3563         DIR="." # File at root of workspace
3564     fi

3566     # Avoid processing the same file twice.
3567     # It's possible for renamed files to
3568     # appear twice in the file list

3570     F=$WDIR/$P

3572     print "<p>"

3574     # If there's a diffs file, make diffs links

3576     if [[ -f $F.cdifff.html ]]; then
3577         cdiff_url="$(print $P.cdifff.html | url_encode)"
3578         udiff_url="$(print $P.udifff.html | url_encode)"

```

```

3579     print "<a href=\"\$cdiff_url\">Cdifff</a>"
3580     print "<a href=\"\$udiff_url\">Udifff</a>"

3582     if [[ -f $F.wdifff.html && -x $WDIFF ]]; then
3583         wdiff_url="$(print $P.wdifff.html | url_encode)"
3584         print "<a href=\"\$wdiff_url\">Wdifff</a>"
3585     fi

3587     sdiff_url="$(print $P.sdiff.html | url_encode)"
3588     print "<a href=\"\$sdiff_url\">Sdifff</a>"

3590     frames_url="$(print $P.frames.html | url_encode)"
3591     print "<a href=\"\$frames_url\">Frames</a>"
3592 else
3593     print " -----"

3595     if [[ -x $WDIFF ]]; then
3596         print " -----"
3597     fi

3599     print " -----"
3600 fi

3602 # If there's an old file, make the link

3604 if [[ -f $F-.html ]]; then
3605     oldfile_url="$(print $P-.html | url_encode)"
3606     print "<a href=\"\$oldfile_url\">Old</a>"
3607 else
3608     print " ---"
3609 fi

3611 # If there's an new file, make the link

3613 if [[ -f $F.html ]]; then
3614     newfile_url="$(print $P.html | url_encode)"
3615     print "<a href=\"\$newfile_url\">New</a>"
3616 else
3617     print " ---"
3618 fi

3620 if [[ -f $F.patch ]]; then
3621     patch_url="$(print $P.patch | url_encode)"
3622     print "<a href=\"\$patch_url\">Patch</a>"
3623 else
3624     print " -----"
3625 fi

3627 if [[ -f $WDIR/raw_files/new/$P ]]; then
3628     rawfiles_url="$(print raw_files/new/$P | url_encode)"
3629     print "<a href=\"\$rawfiles_url\">Raw</a>"
3630 else
3631     print " ---"
3632 fi

3634 print "<b>$P</b>"

3636 # For renamed files, clearly state whether or not they are modified
3637 if [[ -f "$oldname" ]]; then
3638     if [[ -n "$mv_but_nodiff" ]]; then
3639         print "<i>(copied from $oldname)</i>"
3640     else
3641         print "<i>(copied and modified from $oldname)</i>"
3642     fi
3643 elif [[ -n "$oldname" ]]; then
3644     if [[ -n "$mv_but_nodiff" ]]; then

```

```

3645         print "<i>(renamed from $oldname)</i>"
3646     else
3647         print "<i>(renamed and modified from $oldname)</i>"
3648     fi
3649 fi

3651 # If there's an old file, but no new file, the file was deleted
3652 if [[ -f $F-.html && ! -f $F.html ]]; then
3653     print " <i>(deleted)</i>"
3654 fi

3656 #
3657 # Check for usr/closed and deleted_files/usr/closed
3658 #
3659 if [ ! -z "$Oflag" ]; then
3660     if [[ $P == usr/closed/* || \
3661         $P == deleted_files/usr/closed/* ]]; then
3662         print "&nbsp;&nbsp;&nbsp;<i>Closed source: omitted from \
3663             "this review</i>"
3664     fi
3665 fi

3667 print "</p>"
3668 # Insert delta comments

3670 print "<blockquote><pre>"
3671 getcomments html $P $PP
3672 print "</pre>"

3674 # Add additional comments comment

3676 print "<!-- Add comments to explain changes in $P here -->"

3678 # Add count of changes.

3680 if [[ -f $F.count ]]; then
3681     cat $F.count
3682     rm $F.count
3683 fi

3685 if [[ $SCM_MODE == "teamware" ||
3686     $SCM_MODE == "mercurial" ||
3687     $SCM_MODE == "unknown" ]]; then

3689     # Include warnings for important file mode situations:
3690     # 1) New executable files
3691     # 2) Permission changes of any kind
3692     # 3) Existing executable files

3694     old_mode=
3695     if [[ -f $WDIR/raw_files/old/$PP ]]; then
3696         old_mode=`get_file_mode $WDIR/raw_files/old/$PP`
3697     fi

3699     new_mode=
3700     if [[ -f $WDIR/raw_files/new/$P ]]; then
3701         new_mode=`get_file_mode $WDIR/raw_files/new/$P`
3702     fi

3704     if [[ -z "$old_mode" && "$new_mode" = *[1357]* ]]; then
3705         print "<span class=\"chmod\">"
3706         print "<p>new executable file: mode $new_mode</p>"
3707         print "</span>"
3708     elif [[ -n "$old_mode" && -n "$new_mode" &&
3709         "$old_mode" != "$new_mode" ]]; then
3710         print "<span class=\"chmod\">"

```

```

3711         print "<p>mode change: $old_mode to $new_mode</p>"
3712         print "</span>"
3713     elif [[ "$new_mode" = *[1357]* ]]; then
3714         print "<span class=\"chmod\">"
3715         print "<p>executable file: mode $new_mode</p>"
3716         print "</span>"
3717     fi
3718 fi

3720     print "</blockquote>"
3721 done

3723 print
3724 print
3725 print "<hr></hr>"
3726 print "<p style=\"font-size: small\">"
3727 print "This code review page was prepared using <b>$0</b>."
3728 print "Webrev is maintained by the <a href=\"http://www.illumos.org\">"
3729 print "illumos</a> project. The latest version may be obtained"
3730 print "<a href=\"http://src.illumos.org/source/xref/illumos-gate/usr/src/tools/s"
3731 print "</body>"
3732 print "</html>"

3734 exec 1<&- # Close FD 1.
3735 exec 1<&3 # dup FD 3 to restore stdout.
3736 exec 3<&- # close FD 3.

3738 print "Done."

3740 #
3741 # If remote deletion was specified and fails do not continue.
3742 #
3743 if [[ -n $Dflag ]]; then
3744     delete_webrev 1 1
3745     (( $? == 0 )) || exit $?
3746 fi

3748 if [[ -n $Uflag ]]; then
3749     upload_webrev
3750     exit $?
3751 fi

```

new/usr/src/ucbcmd/sbcp/mksysent

1

1908 Sun May 26 18:24:00 2013

new/usr/src/ucbcmd/sbcp/mksysent

XXX Remove awk(1)

```
1 #!/bin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License, Version 1.0 only
7 # (the "License"). You may not use this file except in compliance
8 # with the License.
9 #
10 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
11 # or http://www.opensolaris.org/os/licensing.
12 # See the License for the specific language governing permissions
13 # and limitations under the License.
14 #
15 # When distributing Covered Code, include this CDDL HEADER in each
16 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
17 # If applicable, add the following below this CDDL HEADER, with the
18 # fields enclosed by brackets "[]" replaced with your own identifying
19 # information: Portions Copyright [yyyy] [name of copyright owner]
20 #
21 # CDDL HEADER END
22 #
23 #
24 # Copyright 1993-2003 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 #pragma ident "%Z%M% %I% %E% SMI"
28 #
29 # mksysent - generate the sysent table from the list of system calls
30 #
31 /usr/xpg4/bin/awk '
32 awk '
33 BEGIN {
34     print "#include <sys/asm_linkage.h>"
35     print "#include <sys/syscall.h>"
36     print ""
37     print "! This file is automatically generated from syscalls.list."
38     print "! Do not edit by hand or your changes will be lost."
39     print "! See mksysent and syscalls.list for details."
40     print ""
41     print ".section      \".text\"
42     print ".align 4"
43     print "sysent:"
44 }
```

unchanged_portion_omitted

```

*****
73583 Sun May 26 18:24:01 2013
new/usr/src/uts/common/Makefile.rules
XXX Remove awk(1)
*****
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) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2012 Nexenta Systems, Inc. All rights reserved.
25 #
26 #
27 #
28 # uts/common/Makefile.rules
29 #
30 # This Makefile defines all the file build rules for the directory
31 # uts/common and its children. These are the source files which may
32 # be considered common to all SunOS systems.
33 #
34 # The following two-level ordering must be maintained in this file.
35 # Lines are sorted first in order of decreasing specificity based on
36 # the first directory component. That is, sun4u rules come before
37 # sparc rules come before common rules.
38 #
39 # Lines whose initial directory components are equal are sorted
40 # alphabetically by the remaining components.
41 #
42 #
43 # Section 1a: C objects build rules
44 #
45 $(OBJSDIR)/%.o: $(COMMONBASE)/crypto/aes/%.c
46 $(COMPILE.c) -o $@ $<
47 $(CTFCONVERT_O)
48 #
49 $(OBJSDIR)/%.o: $(COMMONBASE)/crypto/arcfour/%.c
50 $(COMPILE.c) -o $@ $<
51 $(CTFCONVERT_O)
52 #
53 $(OBJSDIR)/%.o: $(COMMONBASE)/crypto/blowfish/%.c
54 $(COMPILE.c) -o $@ $<
55 $(CTFCONVERT_O)
56 #
57 $(OBJSDIR)/%.o: $(COMMONBASE)/crypto/ecc/%.c
58 $(COMPILE.c) -o $@ $<
59 $(CTFCONVERT_O)
60 #
61 $(OBJSDIR)/%.o: $(COMMONBASE)/crypto/modes/%.c

```

```

62 $(COMPILE.c) -o $@ $<
63 $(CTFCONVERT_O)
64 #
65 $(OBJSDIR)/%.o: $(COMMONBASE)/crypto/padding/%.c
66 $(COMPILE.c) -o $@ $<
67 $(CTFCONVERT_O)
68 #
69 $(OBJSDIR)/%.o: $(COMMONBASE)/crypto/rng/%.c
70 $(COMPILE.c) -o $@ $<
71 $(CTFCONVERT_O)
72 #
73 $(OBJSDIR)/%.o: $(COMMONBASE)/crypto/rsa/%.c
74 $(COMPILE.c) -o $@ $<
75 $(CTFCONVERT_O)
76 #
77 $(OBJSDIR)/%.o: $(COMMONBASE)/bignum/%.c
78 $(COMPILE.c) -o $@ $<
79 $(CTFCONVERT_O)
80 #
81 $(OBJSDIR)/%.o: $(UTSBASE)/common/bignum/%.c
82 $(COMPILE.c) -o $@ $<
83 $(CTFCONVERT_O)
84 #
85 $(OBJSDIR)/%.o: $(COMMONBASE)/mpi/%.c
86 $(COMPILE.c) -o $@ $<
87 $(CTFCONVERT_O)
88 #
89 $(OBJSDIR)/%.o: $(COMMONBASE)/acl/%.c
90 $(COMPILE.c) -o $@ $<
91 $(CTFCONVERT_O)
92 #
93 $(OBJSDIR)/%.o: $(COMMONBASE)/avl/%.c
94 $(COMPILE.c) -o $@ $<
95 $(CTFCONVERT_O)
96 #
97 $(OBJSDIR)/%.o: $(COMMONBASE)/ucode/%.c
98 $(COMPILE.c) -o $@ $<
99 $(CTFCONVERT_O)
100 #
101 $(OBJSDIR)/%.o: $(UTSBASE)/common/brand/sn1/%.c
102 $(COMPILE.c) -o $@ $<
103 $(CTFCONVERT_O)
104 #
105 $(OBJSDIR)/%.o: $(UTSBASE)/common/brand/solaris10/%.c
106 $(COMPILE.c) -o $@ $<
107 $(CTFCONVERT_O)
108 #
109 $(OBJSDIR)/%.o: $(UTSBASE)/common/c2/%.c
110 $(COMPILE.c) -o $@ $<
111 $(CTFCONVERT_O)
112 #
113 $(OBJSDIR)/%.o: $(UTSBASE)/common/conf/%.c
114 $(COMPILE.c) -o $@ $<
115 $(CTFCONVERT_O)
116 #
117 $(OBJSDIR)/%.o: $(UTSBASE)/common/contract/%.c
118 $(COMPILE.c) -o $@ $<
119 $(CTFCONVERT_O)
120 #
121 $(OBJSDIR)/%.o: $(UTSBASE)/common/cpr/%.c
122 $(COMPILE.c) -o $@ $<
123 $(CTFCONVERT_O)
124 #
125 $(OBJSDIR)/%.o: $(UTSBASE)/common/ctf/%.c
126 $(COMPILE.c) -o $@ $<
127 $(CTFCONVERT_O)

```

```

129 $(OBJSDIR)/%.o: $(COMMONBASE)/ctf/%.c
130 $(COMPILE.c) -o $@ $<
131 $(CTFCONVERT_O)

133 $(OBJSDIR)/%.o: $(COMMONBASE)/crypto/des/%.c
134 $(COMPILE.c) -o $@ $<
135 $(CTFCONVERT_O)

137 $(OBJSDIR)/%.o: $(COMMONBASE)/smbios/%.c
138 $(COMPILE.c) -o $@ $<
139 $(CTFCONVERT_O)

141 $(OBJSDIR)/%.o: $(UTSBASE)/common/des/%.c
142 $(COMPILE.c) -o $@ $<
143 $(CTFCONVERT_O)

145 $(OBJSDIR)/%.o: $(UTSBASE)/common/crypto/api/%.c
146 $(COMPILE.c) -o $@ $<
147 $(CTFCONVERT_O)

149 $(OBJSDIR)/%.o: $(UTSBASE)/common/crypto/core/%.c
150 $(COMPILE.c) -o $@ $<
151 $(CTFCONVERT_O)

153 $(OBJSDIR)/%.o: $(UTSBASE)/common/crypto/io/%.c
154 $(COMPILE.c) -o $@ $<
155 $(CTFCONVERT_O)

157 $(OBJSDIR)/%.o: $(UTSBASE)/common/crypto/spi/%.c
158 $(COMPILE.c) -o $@ $<
159 $(CTFCONVERT_O)

161 $(OBJSDIR)/%.o: $(COMMONBASE)/pci/%.c
162 $(COMPILE.c) -o $@ $<
163 $(CTFCONVERT_O)

165 $(OBJSDIR)/%.o: $(COMMONBASE)/devid/%.c
166 $(COMPILE.c) -o $@ $<
167 $(CTFCONVERT_O)

169 $(OBJSDIR)/%.o: $(UTSBASE)/common/disp/%.c
170 $(COMPILE.c) -o $@ $<
171 $(CTFCONVERT_O)

173 $(OBJSDIR)/%.o: $(UTSBASE)/common/dtrace/%.c
174 $(COMPILE.c) -o $@ $<
175 $(CTFCONVERT_O)

177 $(OBJSDIR)/%.o: $(COMMONBASE)/exacct/%.c
178 $(COMPILE.c) -o $@ $<
179 $(CTFCONVERT_O)

181 $(OBJSDIR)/%.o: $(UTSBASE)/common/exec/aout/%.c
182 $(COMPILE.c) -o $@ $<
183 $(CTFCONVERT_O)

185 $(OBJSDIR)/%.o: $(UTSBASE)/common/exec/elf/%.c
186 $(COMPILE.c) -o $@ $<
187 $(CTFCONVERT_O)

189 $(OBJSDIR)/%.o: $(UTSBASE)/common/exec/intp/%.c
190 $(COMPILE.c) -o $@ $<
191 $(CTFCONVERT_O)

193 $(OBJSDIR)/%.o: $(UTSBASE)/common/exec/shbin/%.c

```

```

194 $(COMPILE.c) -o $@ $<
195 $(CTFCONVERT_O)

197 $(OBJSDIR)/%.o: $(UTSBASE)/common/exec/java/%.c
198 $(COMPILE.c) -o $@ $<
199 $(CTFCONVERT_O)

201 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/%.c
202 $(COMPILE.c) -o $@ $<
203 $(CTFCONVERT_O)

205 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/autofs/%.c
206 $(COMPILE.c) -o $@ $<
207 $(CTFCONVERT_O)

209 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/cacheofs/%.c
210 $(COMPILE.c) -o $@ $<
211 $(CTFCONVERT_O)

213 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/dcfis/%.c
214 $(COMPILE.c) -o $@ $<
215 $(CTFCONVERT_O)

217 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/devfs/%.c
218 $(COMPILE.c) -o $@ $<
219 $(CTFCONVERT_O)

221 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/ctfs/%.c
222 $(COMPILE.c) -o $@ $<
223 $(CTFCONVERT_O)

225 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/doorfs/%.c
226 $(COMPILE.c) -o $@ $<
227 $(CTFCONVERT_O)

229 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/dev/%.c
230 $(COMPILE.c) -o $@ $<
231 $(CTFCONVERT_O)

233 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/fd/%.c
234 $(COMPILE.c) -o $@ $<
235 $(CTFCONVERT_O)

237 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/fifofs/%.c
238 $(COMPILE.c) -o $@ $<
239 $(CTFCONVERT_O)

241 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/hsfs/%.c
242 $(COMPILE.c) -o $@ $<
243 $(CTFCONVERT_O)

245 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/lofs/%.c
246 $(COMPILE.c) -o $@ $<
247 $(CTFCONVERT_O)

249 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/mntfs/%.c
250 $(COMPILE.c) -o $@ $<
251 $(CTFCONVERT_O)

253 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/namefs/%.c
254 $(COMPILE.c) -o $@ $<
255 $(CTFCONVERT_O)

257 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/nfs/%.c
258 $(COMPILE.c) -o $@ $<
259 $(CTFCONVERT_O)

```



```

261 $(OBJSDIR)/%.o: $(COMMONBASE)/smbsrv/%.c
262 $(COMPILE.c) -o $@ $<
263 $(CTFCONVERT_O)

265 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/smbsrv/%.c
266 $(COMPILE.c) -o $@ $<
267 $(CTFCONVERT_O)

269 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/objfs/%.c
270 $(COMPILE.c) -o $@ $<
271 $(CTFCONVERT_O)

273 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/pcfefs/%.c
274 $(COMPILE.c) -o $@ $<
275 $(CTFCONVERT_O)

277 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/portfs/%.c
278 $(COMPILE.c) -o $@ $<
279 $(CTFCONVERT_O)

281 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/proc/%.c
282 $(COMPILE.c) -o $@ $<
283 $(CTFCONVERT_O)

285 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/sharefs/%.c
286 $(COMPILE.c) -o $@ $<
287 $(CTFCONVERT_O)

289 $(OBJSDIR)/%.o: $(COMMONBASE)/smbclnt/%.c
290 $(COMPILE.c) -o $@ $<
291 $(CTFCONVERT_O)

293 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/smbclnt/netsmb/%.c
294 $(COMPILE.c) -o $@ $<
295 $(CTFCONVERT_O)

297 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/smbclnt/smbfs/%.c
298 $(COMPILE.c) -o $@ $<
299 $(CTFCONVERT_O)

301 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/sockfs/%.c
302 $(COMPILE.c) -o $@ $<
303 $(CTFCONVERT_O)

305 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/specfs/%.c
306 $(COMPILE.c) -o $@ $<
307 $(CTFCONVERT_O)

309 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/swapfs/%.c
310 $(COMPILE.c) -o $@ $<
311 $(CTFCONVERT_O)

313 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/tmpfs/%.c
314 $(COMPILE.c) -o $@ $<
315 $(CTFCONVERT_O)

317 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/udfs/%.c
318 $(COMPILE.c) -o $@ $<
319 $(CTFCONVERT_O)

321 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/ufs/%.c
322 $(COMPILE.c) -o $@ $<
323 $(CTFCONVERT_O)

325 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/vscan/%.c

```

```

326 $(COMPILE.c) -o $@ $<
327 $(CTFCONVERT_O)

329 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/zfs/%.c
330 $(COMPILE.c) -o $@ $<
331 $(CTFCONVERT_O)

333 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/zut/%.c
334 $(COMPILE.c) -o $@ $<
335 $(CTFCONVERT_O)

337 $(OBJSDIR)/%.o: $(COMMONBASE)/xattr/%.c
338 $(COMPILE.c) -o $@ $<
339 $(CTFCONVERT_O)

341 $(OBJSDIR)/%.o: $(COMMONBASE)/zfs/%.c
342 $(COMPILE.c) -o $@ $<
343 $(CTFCONVERT_O)

345 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/scsi/adapters/pmcs/%.c
346 $(COMPILE.c) -o $@ $<
347 $(CTFCONVERT_O)

349 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/scsi/adapters/pmcs/%.bin
350 $(COMPILE.b) -o $@ $<
351 $(CTFCONVERT_O)

353 $(OBJSDIR)/%.o: $(COMMONBASE)/fsreparse/%.c
354 $(COMPILE.c) -o $@ $<
355 $(CTFCONVERT_O)

357 KMECHKRB5_BASE=$(UTSBASE)/common/gssapi/mechs/krb5

359 KGSSDFLAGS=-I $(UTSBASE)/common/gssapi/include

361 # Note, KRB5_DEFS can be assigned various preprocessor flags,
362 # typically -D defines on the make invocation. The standard compiler
363 # flags will not be overwritten.
364 KGSSDFLAGS += $(KRB5_DEFS)

366 $(OBJSDIR)/%.o: $(UTSBASE)/common/gssapi/%.c
367 $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
368 $(CTFCONVERT_O)

370 $(OBJSDIR)/%.o: $(UTSBASE)/common/gssapi/mechs/dummy/%.c
371 $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
372 $(CTFCONVERT_O)

374 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/%.c
375 $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
376 $(CTFCONVERT_O)

378 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/crypto/%.c
379 $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
380 $(CTFCONVERT_O)

382 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/crypto/des/%.c
383 $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
384 $(CTFCONVERT_O)

386 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/crypto/arcfour/%.c
387 $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
388 $(CTFCONVERT_O)

390 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/crypto/dk/%.c
391 $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<

```

```

392      $(CTFCONVERT_O)

394 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/enc_provider/%.c
395     $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
396     $(CTFCONVERT_O)

398 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/hash_provider/%.c
399     $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
400     $(CTFCONVERT_O)

402 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/keyhash_provider/%.c
403     $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
404     $(CTFCONVERT_O)

406 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/raw/%.c
407     $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
408     $(CTFCONVERT_O)

410 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/old/%.c
411     $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
412     $(CTFCONVERT_O)

414 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/krb5/krb/%.c
415     $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
416     $(CTFCONVERT_O)

418 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/krb5/os/%.c
419     $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
420     $(CTFCONVERT_O)

422 $(OBJS_DIR)/ser_sctx.o := CPPFLAGS += -DPROVIDE_KERNEL_IMPORT=1

424 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/mech/%.c
425     $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
426     $(CTFCONVERT_O)

428 $(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/profile/%.c
429     $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
430     $(CTFCONVERT_O)

432 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ncall/%.c
433     $(COMPILE.c) -o $@ $<
434     $(CTFCONVERT_O)

436 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/dsw/%.c
437     $(COMPILE.c) -o $@ $<
438     $(CTFCONVERT_O)

440 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/nsctl/%.c
441     $(COMPILE.c) -o $@ $<
442     $(CTFCONVERT_O)

444 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/rdc/%.c
445     $(COMPILE.c) -o $@ $<
446     $(CTFCONVERT_O)

448 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/sdbc/%.c
449     $(COMPILE.c) -o $@ $<
450     $(CTFCONVERT_O)

452 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/solaris/%.c
453     $(COMPILE.c) -o $@ $<
454     $(CTFCONVERT_O)

456 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/sv/%.c
457     $(COMPILE.c) -o $@ $<

```

```

458      $(CTFCONVERT_O)

460 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/avs/ns/unistat/%.c
461     $(COMPILE.c) -o $@ $<
462     $(CTFCONVERT_O)

464 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/idmap/%.c
465     $(COMPILE.c) -o $@ $<
466     $(CTFCONVERT_O)

468 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/%.c
469     $(COMPILE.c) -o $@ $<
470     $(CTFCONVERT_O)

472 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/arp/%.c
473     $(COMPILE.c) -o $@ $<
474     $(CTFCONVERT_O)

476 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/ip/%.c
477     $(COMPILE.c) -o $@ $<
478     $(CTFCONVERT_O)

480 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/ipnet/%.c
481     $(COMPILE.c) -o $@ $<
482     $(CTFCONVERT_O)

484 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/iptun/%.c
485     $(COMPILE.c) -o $@ $<
486     $(CTFCONVERT_O)

488 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/kssl/%.c
489     $(COMPILE.c) -o $@ $<
490     $(CTFCONVERT_O)

492 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/sctp/%.c
493     $(COMPILE.c) -o $@ $<
494     $(CTFCONVERT_O)

496 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/tcp/%.c
497     $(COMPILE.c) -o $@ $<
498     $(CTFCONVERT_O)

500 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/ilb/%.c
501     $(COMPILE.c) -o $@ $<
502     $(CTFCONVERT_O)

504 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/ipf/%.c
505     $(COMPILE.c) -o $@ $<
506     $(CTFCONVERT_O)

508 $(OBJS_DIR)/%.o:                $(COMMONBASE)/net/patricia/%.c
509     $(COMPILE.c) -o $@ $<
510     $(CTFCONVERT_O)

512 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/udp/%.c
513     $(COMPILE.c) -o $@ $<
514     $(CTFCONVERT_O)

516 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/nca/%.c
517     $(COMPILE.c) -o $@ $<
518     $(CTFCONVERT_O)

520 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/sockmods/%.c
521     $(COMPILE.c) -o $@ $<
522     $(CTFCONVERT_O)

```

```

524 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/dlpistub/%.c
525 $(COMPILE.c) -o $@ $<
526 $(CTFCONVERT_O)

528 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/%.c
529 $(COMPILE.c) -o $@ $<
530 $(CTFCONVERT_O)

532 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/l394/%.c
533 $(COMPILE.c) -o $@ $<
534 $(CTFCONVERT_O)

536 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/l394/adapters/%.c
537 $(COMPILE.c) -o $@ $<
538 $(CTFCONVERT_O)

540 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/l394/targets/av1394/%.c
541 $(COMPILE.c) -o $@ $<
542 $(CTFCONVERT_O)

544 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/l394/targets/dcam1394/%.c
545 $(COMPILE.c) -o $@ $<
546 $(CTFCONVERT_O)

548 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/l394/targets/scsal394/%.c
549 $(COMPILE.c) -o $@ $<
550 $(CTFCONVERT_O)

552 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/sbp2/%.c
553 $(COMPILE.c) -o $@ $<
554 $(CTFCONVERT_O)

556 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/aac/%.c
557 $(COMPILE.c) -o $@ $<
558 $(CTFCONVERT_O)

560 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/afe/%.c
561 $(COMPILE.c) -o $@ $<
562 $(CTFCONVERT_O)

564 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/atge/%.c
565 $(COMPILE.c) -o $@ $<
566 $(CTFCONVERT_O)

568 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/arn/%.c
569 $(COMPILE.c) -o $@ $<
570 $(CTFCONVERT_O)

572 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ath/%.c
573 $(COMPILE.c) -o $@ $<
574 $(CTFCONVERT_O)

576 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/atu/%.c
577 $(COMPILE.c) -o $@ $<
578 $(CTFCONVERT_O)

580 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/impl/%.c
581 $(COMPILE.c) -o $@ $<
582 $(CTFCONVERT_O)

584 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/ac97/%.c
585 $(COMPILE.c) -o $@ $<
586 $(CTFCONVERT_O)

588 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audioens/%.c
589 $(COMPILE.c) -o $@ $<

```

```

590 $(CTFCONVERT_O)

592 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audioemu10k/%.c
593 $(COMPILE.c) -o $@ $<
594 $(CTFCONVERT_O)

596 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audio1575/%.c
597 $(COMPILE.c) -o $@ $<
598 $(CTFCONVERT_O)

600 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audio810/%.c
601 $(COMPILE.c) -o $@ $<
602 $(CTFCONVERT_O)

604 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audiocmi/%.c
605 $(COMPILE.c) -o $@ $<
606 $(CTFCONVERT_O)

608 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audiocmihd/%.c
609 $(COMPILE.c) -o $@ $<
610 $(CTFCONVERT_O)

612 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audiohd/%.c
613 $(COMPILE.c) -o $@ $<
614 $(CTFCONVERT_O)

616 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audioixp/%.c
617 $(COMPILE.c) -o $@ $<
618 $(CTFCONVERT_O)

620 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audiols/%.c
621 $(COMPILE.c) -o $@ $<
622 $(CTFCONVERT_O)

624 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audiopci/%.c
625 $(COMPILE.c) -o $@ $<
626 $(CTFCONVERT_O)

628 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audiopl6x/%.c
629 $(COMPILE.c) -o $@ $<
630 $(CTFCONVERT_O)

632 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audiosolo/%.c
633 $(COMPILE.c) -o $@ $<
634 $(CTFCONVERT_O)

636 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audiots/%.c
637 $(COMPILE.c) -o $@ $<
638 $(CTFCONVERT_O)

640 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audiovia823x/%.c
641 $(COMPILE.c) -o $@ $<
642 $(CTFCONVERT_O)

644 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/audio/drv/audiovia97/%.c
645 $(COMPILE.c) -o $@ $<
646 $(CTFCONVERT_O)

648 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/bfe/%.c
649 $(COMPILE.c) -o $@ $<
650 $(CTFCONVERT_O)

652 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/bge/%.c
653 $(COMPILE.c) -o $@ $<
654 $(CTFCONVERT_O)

```

```

656 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/blkdev/%.c
657 $(COMPILE.c) -o $@ $<
658 $(CTFCONVERT_O)

660 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/bpf/%.c
661 $(COMPILE.c) -o $@ $<
662 $(CTFCONVERT_O)

664 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/cardbus/%.c
665 $(COMPILE.c) -o $@ $<
666 $(CTFCONVERT_O)

668 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/stmf/%.c
669 $(COMPILE.c) -o $@ $<
670 $(CTFCONVERT_O)

672 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/port/fct/%.c
673 $(COMPILE.c) -o $@ $<
674 $(CTFCONVERT_O)

676 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/port/qlt/%.c
677 $(COMPILE.c) -o $@ $<
678 $(CTFCONVERT_O)

680 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/port/srpt/%.c
681 $(COMPILE.c) -o $@ $<
682 $(CTFCONVERT_O)

684 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/port/fcoet/%.c
685 $(COMPILE.c) -o $@ $<
686 $(CTFCONVERT_O)

688 $(OBJSDIR)/%.o: $(COMMONBASE)/iscsit/%.c
689 $(COMPILE.c) -o $@ $<
690 $(CTFCONVERT_O)

692 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/port/iscsit/%.c
693 $(COMPILE.c) -o $@ $<
694 $(CTFCONVERT_O)

696 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/port/pppt/%.c
697 $(COMPILE.c) -o $@ $<
698 $(CTFCONVERT_O)

700 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/lu/stmf_sbd/%.c
701 $(COMPILE.c) -o $@ $<
702 $(CTFCONVERT_O)

704 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/dld/%.c
705 $(COMPILE.c) -o $@ $<
706 $(CTFCONVERT_O)

708 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/dls/%.c
709 $(COMPILE.c) -o $@ $<
710 $(CTFCONVERT_O)

712 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/dmfe/%.c
713 $(COMPILE.c) -o $@ $<
714 $(CTFCONVERT_O)

716 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/drm/%.c
717 $(COMPILE.c) -o $@ $<
718 $(CTFCONVERT_O)

720 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/efe/%.c
721 $(COMPILE.c) -o $@ $<

```

```

722 $(CTFCONVERT_O)

724 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/elxl/%.c
725 $(COMPILE.c) -o $@ $<
726 $(CTFCONVERT_O)

728 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/fcoe/%.c
729 $(COMPILE.c) -o $@ $<
730 $(CTFCONVERT_O)

732 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/hme/%.c
733 $(COMPILE.c) -o $@ $<
734 $(CTFCONVERT_O)

736 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/pciex/%.c
737 $(COMPILE.c) -o $@ $<
738 $(CTFCONVERT_O)

740 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/hotplug/hpcsvc/%.c
741 $(COMPILE.c) -o $@ $<
742 $(CTFCONVERT_O)

744 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/pciex/hotplug/%.c
745 $(COMPILE.c) -o $@ $<
746 $(CTFCONVERT_O)

748 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/hotplug/pcihp/%.c
749 $(COMPILE.c) -o $@ $<
750 $(CTFCONVERT_O)

752 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/rds/%.c
753 $(COMPILE.c) -o $@ $<
754 $(CTFCONVERT_O)

756 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/rdsv3/%.c
757 $(COMPILE.c) -o $@ $<
758 $(CTFCONVERT_O)

760 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/iser/%.c
761 $(COMPILE.c) -o $@ $<
762 $(CTFCONVERT_O)

764 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/ibd/%.c
765 $(COMPILE.c) -o $@ $<
766 $(CTFCONVERT_O)

768 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/eoib/%.c
769 $(COMPILE.c) -o $@ $<
770 $(CTFCONVERT_O)

772 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/of/sol_ofs/%.c
773 $(COMPILE.c) -o $@ $<
774 $(CTFCONVERT_O)

776 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/of/sol_ucma/%.c
777 $(COMPILE.c) -o $@ $<
778 $(CTFCONVERT_O)

780 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/of/sol_umad/%.c
781 $(COMPILE.c) -o $@ $<
782 $(CTFCONVERT_O)

784 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/of/sol_uverbs/%.c
785 $(COMPILE.c) -o $@ $<
786 $(CTFCONVERT_O)

```

```

788 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/sdp/%.c
789 $(COMPILE.c) -o $@ $<
790 $(CTFCONVERT_O)

792 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/mgt/ibcm/%.c
793 $(COMPILE.c) -o $@ $<
794 $(CTFCONVERT_O)

796 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/mgt/ibdm/%.c
797 $(COMPILE.c) -o $@ $<
798 $(CTFCONVERT_O)

800 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/mgt/ibdma/%.c
801 $(COMPILE.c) -o $@ $<
802 $(CTFCONVERT_O)

804 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/mgt/ibmf/%.c
805 $(COMPILE.c) -o $@ $<
806 $(CTFCONVERT_O)

808 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/ibnex/%.c
809 $(COMPILE.c) -o $@ $<
810 $(CTFCONVERT_O)

812 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/ibtl/%.c
813 $(COMPILE.c) -o $@ $<
814 $(CTFCONVERT_O)

816 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/adapters/tavor/%.c
817 $(COMPILE.c) -o $@ $<
818 $(CTFCONVERT_O)

820 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/adapters/hermon/%.c
821 $(COMPILE.c) -o $@ $<
822 $(CTFCONVERT_O)

824 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/daplt/%.c
825 $(COMPILE.c) -o $@ $<
826 $(CTFCONVERT_O)

828 $(OBJSDIR)/%.o: $(COMMONBASE)/iscsi/%.c
829 $(COMPILE.c) -o $@ $<
830 $(CTFCONVERT_O)

832 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/idm/%.c
833 $(COMPILE.c) -o $@ $<
834 $(CTFCONVERT_O)

836 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ipw/%.c
837 $(COMPILE.c) -o $@ $<
838 $(CTFCONVERT_O)

840 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/iwh/%.c
841 $(COMPILE.c) -o $@ $<
842 $(CTFCONVERT_O)

844 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/iwi/%.c
845 $(COMPILE.c) -o $@ $<
846 $(CTFCONVERT_O)

848 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/iwk/%.c
849 $(COMPILE.c) -o $@ $<
850 $(CTFCONVERT_O)

852 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/iwp/%.c
853 $(COMPILE.c) -o $@ $<

```

```

854 $(CTFCONVERT_O)

856 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/kb8042/%.c
857 $(COMPILE.c) -o $@ $<
858 $(CTFCONVERT_O)

860 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/kbtrans/%.c
861 $(COMPILE.c) -o $@ $<
862 $(CTFCONVERT_O)

864 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ksocket/%.c
865 $(COMPILE.c) -o $@ $<
866 $(CTFCONVERT_O)

868 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/aggr/%.c
869 $(COMPILE.c) -o $@ $<
870 $(CTFCONVERT_O)

872 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/lp/%.c
873 $(COMPILE.c) -o $@ $<
874 $(CTFCONVERT_O)

876 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/lvm/hotspares/%.c
877 $(COMPILE.c) -o $@ $<
878 $(CTFCONVERT_O)

880 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/lvm/md/%.c
881 $(COMPILE.c) -o $@ $<
882 $(CTFCONVERT_O)

884 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/lvm/mirror/%.c
885 $(COMPILE.c) -o $@ $<
886 $(CTFCONVERT_O)

888 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/lvm/notify/%.c
889 $(COMPILE.c) -o $@ $<
890 $(CTFCONVERT_O)

892 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/lvm/raid/%.c
893 $(COMPILE.c) -o $@ $<
894 $(CTFCONVERT_O)

896 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/lvm/softpart/%.c
897 $(COMPILE.c) -o $@ $<
898 $(CTFCONVERT_O)

900 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/lvm/striped/%.c
901 $(COMPILE.c) -o $@ $<
902 $(CTFCONVERT_O)

904 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/lvm/trans/%.c
905 $(COMPILE.c) -o $@ $<
906 $(CTFCONVERT_O)

908 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/mac/%.c
909 $(COMPILE.c) -o $@ $<
910 $(CTFCONVERT_O)

912 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/mac/plugins/%.c
913 $(COMPILE.c) -o $@ $<
914 $(CTFCONVERT_O)

916 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/mega_sas/%.c
917 $(COMPILE.c) -o $@ $<
918 $(CTFCONVERT_O)

```

```

920 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/mii/%.c
921     $(COMPILE.c) -o $@ $<
922     $(CTFCONVERT_O)

924 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/mr_sas/%.c
925     $(COMPILE.c) -o $@ $<
926     $(CTFCONVERT_O)

928 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/scsi/adapters/mpt_sas/%.c
929     $(COMPILE.c) -o $@ $<
930     $(CTFCONVERT_O)

932 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/mxfe/%.c
933     $(COMPILE.c) -o $@ $<
934     $(CTFCONVERT_O)

936 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/mwl/%.c
937     $(COMPILE.c) -o $@ $<
938     $(CTFCONVERT_O)

940 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/mwl/mwl_fw/%.c
941     $(COMPILE.c) -o $@ $<
942     $(CTFCONVERT_O)

944 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/net80211/%.c
945     $(COMPILE.c) -o $@ $<
946     $(CTFCONVERT_O)

948 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/nge/%.c
949     $(COMPILE.c) -o $@ $<
950     $(CTFCONVERT_O)

952 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/nxge/%.c
953     $(COMPILE.c) -o $@ $<
954     $(CTFCONVERT_O)

956 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/nxge/npi/%.c
957     $(COMPILE.c) -o $@ $<
958     $(CTFCONVERT_O)

960 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/nxge/%.s
961     $(COMPILE.s) -o $@ $<

963 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/pci-ide/%.c
964     $(COMPILE.c) -o $@ $<
965     $(CTFCONVERT_O)

967 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/pcmcia/%.c
968     $(COMPILE.c) -o $@ $<
969     $(CTFCONVERT_O)

971 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/pcan/%.c
972     $(COMPILE.c) -o $@ $<
973     $(CTFCONVERT_O)

975 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/pcn/%.c
976     $(COMPILE.c) -o $@ $<
977     $(CTFCONVERT_O)

979 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/pcwl/%.c
980     $(COMPILE.c) -o $@ $<
981     $(CTFCONVERT_O)

983 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ppp/sppp/%.c
984     $(COMPILE.c) -o $@ $<
985     $(CTFCONVERT_O)

```

```

987 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ppp/sppasyn/%.c
988     $(COMPILE.c) -o $@ $<
989     $(CTFCONVERT_O)

991 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ppp/sppptun/%.c
992     $(COMPILE.c) -o $@ $<
993     $(CTFCONVERT_O)

995 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ral/%.c
996     $(COMPILE.c) -o $@ $<
997     $(CTFCONVERT_O)

999 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/rge/%.c
1000    $(COMPILE.c) -o $@ $<
1001    $(CTFCONVERT_O)

1003 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/rtls/%.c
1004    $(COMPILE.c) -o $@ $<
1005    $(CTFCONVERT_O)

1007 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/rsm/%.c
1008    $(COMPILE.c) -o $@ $<
1009    $(CTFCONVERT_O)

1011 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/rtw/%.c
1012    $(COMPILE.c) -o $@ $<
1013    $(CTFCONVERT_O)

1015 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/rum/%.c
1016    $(COMPILE.c) -o $@ $<
1017    $(CTFCONVERT_O)

1019 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/rwd/%.c
1020    $(COMPILE.c) -o $@ $<
1021    $(CTFCONVERT_O)

1023 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/rwn/%.c
1024    $(COMPILE.c) -o $@ $<
1025    $(CTFCONVERT_O)

1027 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/sata/adapters/ahci/%.c
1028    $(COMPILE.c) -o $@ $<
1029    $(CTFCONVERT_O)

1031 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/sata/adapters/nv_sata/%.c
1032    $(COMPILE.c) -o $@ $<
1033    $(CTFCONVERT_O)

1035 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/sata/adapters/si3124/%.c
1036    $(COMPILE.c) -o $@ $<
1037    $(CTFCONVERT_O)

1039 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/sata/impl/%.c
1040    $(COMPILE.c) -o $@ $<
1041    $(CTFCONVERT_O)

1043 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/scsi/conf/%.c
1044    $(COMPILE.c) -o $@ $<
1045    $(CTFCONVERT_O)

1047 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/scsi/impl/%.c
1048    $(COMPILE.c) -o $@ $<
1049    $(CTFCONVERT_O)

1051 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/scsi/targets/%.c

```

```

1052 $(COMPILE.c) -o $@ $<
1053 $(CTFCONVERT_O)

1055 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/scsi/adapters/%.c
1056 $(COMPILE.c) -o $@ $<
1057 $(CTFCONVERT_O)

1059 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/scsi/adapters/blk2scsa/%.c
1060 $(COMPILE.c) -o $@ $<
1061 $(CTFCONVERT_O)

1063 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/%.c
1064 $(COMPILE.c) -o $@ $<
1065 $(CTFCONVERT_O)

1067 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/fop
1068 $(COMPILE.c) -o $@ $<
1069 $(CTFCONVERT_O)

1071 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/fibre-channel/ulp/%.c
1072 $(COMPILE.c) -o $@ $<
1073 $(CTFCONVERT_O)

1075 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/fibre-channel/impl/%.c
1076 $(COMPILE.c) -o $@ $<
1077 $(CTFCONVERT_O)

1079 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/fibre-channel/fca/qlc/%.c
1080 $(COMPILE.c) -o $@ $<
1081 $(CTFCONVERT_O)

1083 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/fibre-channel/fca/qlge/%.c
1084 $(COMPILE.c) -o $@ $<
1085 $(CTFCONVERT_O)

1087 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/fibre-channel/fca/emlxs/%.c
1088 $(COMPILE.c) -o $@ $<
1089 $(CTFCONVERT_O)

1091 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/fibre-channel/fca/oce/%.c
1092 $(COMPILE.c) -o $@ $<
1093 $(CTFCONVERT_O)

1095 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/fibre-channel/fca/fcoei/%.c
1096 $(COMPILE.c) -o $@ $<
1097 $(CTFCONVERT_O)

1099 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/sdcard/adapters/sdhost/%.c
1100 $(COMPILE.c) -o $@ $<
1101 $(CTFCONVERT_O)

1103 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/sdcard/impl/%.c
1104 $(COMPILE.c) -o $@ $<
1105 $(CTFCONVERT_O)

1107 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/sdcard/targets/sdcard/%.c
1108 $(COMPILE.c) -o $@ $<
1109 $(CTFCONVERT_O)

1111 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/sfe/%.c
1112 $(COMPILE.c) -o $@ $<
1113 $(CTFCONVERT_O)

1115 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/simnet/%.c
1116 $(COMPILE.c) -o $@ $<
1117 $(CTFCONVERT_O)

```

```

1119 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/softmac/%.c
1120 $(COMPILE.c) -o $@ $<
1121 $(CTFCONVERT_O)

1123 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/uath/%.c
1124 $(COMPILE.c) -o $@ $<
1125 $(CTFCONVERT_O)

1127 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/uath/uath_fw/%.c
1128 $(COMPILE.c) -o $@ $<
1129 $(CTFCONVERT_O)

1131 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ural/%.c
1132 $(COMPILE.c) -o $@ $<
1133 $(CTFCONVERT_O)

1135 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/urtw/%.c
1136 $(COMPILE.c) -o $@ $<
1137 $(CTFCONVERT_O)

1139 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/audio/usb_ac/%.c
1140 $(COMPILE.c) -o $@ $<
1141 $(CTFCONVERT_O)

1143 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/audio/usb_as/%.c
1144 $(COMPILE.c) -o $@ $<
1145 $(CTFCONVERT_O)

1147 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/audio/usb_ah/%.c
1148 $(COMPILE.c) -o $@ $<
1149 $(CTFCONVERT_O)

1151 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/usbskel/%.c
1152 $(COMPILE.c) -o $@ $<
1153 $(CTFCONVERT_O)

1155 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/video/usbvc/%.c
1156 $(COMPILE.c) -o $@ $<
1157 $(CTFCONVERT_O)

1159 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/hwarc/%.c
1160 $(COMPILE.c) -o $@ $<
1161 $(CTFCONVERT_O)

1163 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/hid/%.c
1164 $(COMPILE.c) -o $@ $<
1165 $(CTFCONVERT_O)

1167 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/hidparser/%.c
1168 $(COMPILE.c) -o $@ $<
1169 $(CTFCONVERT_O)

1171 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/printer/%.c
1172 $(COMPILE.c) -o $@ $<
1173 $(CTFCONVERT_O)

1175 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/usbbkm/%.c
1176 $(COMPILE.c) -o $@ $<
1177 $(CTFCONVERT_O)

1179 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/uslbs/%.c
1180 $(COMPILE.c) -o $@ $<
1181 $(CTFCONVERT_O)

1183 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/usbinp/usbwcm

```

```

1184 $(COMPILE.c) -o $$@ $<
1185 $(CTFCONVERT_O)

1187 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/ugen/%.c
1188 $(COMPILE.c) -o $$@ $<
1189 $(CTFCONVERT_O)

1191 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/usbser/%.c
1192 $(COMPILE.c) -o $$@ $<
1193 $(CTFCONVERT_O)

1195 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/usbser/usbsacm/
1196 $(COMPILE.c) -o $$@ $<
1197 $(CTFCONVERT_O)

1199 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/usbser/usbftdi/
1200 $(COMPILE.c) -o $$@ $<
1201 $(CTFCONVERT_O)

1203 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/usbser/usbser_k
1204 $(COMPILE.c) -o $$@ $<
1205 $(CTFCONVERT_O)

1207 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/usbser/usbsprl/
1208 $(COMPILE.c) -o $$@ $<
1209 $(CTFCONVERT_O)

1211 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/wusb_df/%.c
1212 $(COMPILE.c) -o $$@ $<
1213 $(CTFCONVERT_O)

1215 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/hwal480_fw/%.c
1216 $(COMPILE.c) -o $$@ $<
1217 $(CTFCONVERT_O)

1219 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/wusb_ca/%.c
1220 $(COMPILE.c) -o $$@ $<
1221 $(CTFCONVERT_O)

1223 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/clients/usbecm/%.c
1224 $(COMPILE.c) -o $$@ $<
1225 $(CTFCONVERT_O)

1227 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/hcd/openhci/%.c
1228 $(COMPILE.c) -o $$@ $<
1229 $(CTFCONVERT_O)

1231 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/hcd/ehci/%.c
1232 $(COMPILE.c) -o $$@ $<
1233 $(CTFCONVERT_O)

1235 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/hcd/uhci/%.c
1236 $(COMPILE.c) -I../common -o $$@ $<
1237 $(CTFCONVERT_O)

1239 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/hubd/%.c
1240 $(COMPILE.c) -o $$@ $<
1241 $(CTFCONVERT_O)

1243 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/scsa2usb/%.c
1244 $(COMPILE.c) -o $$@ $<
1245 $(CTFCONVERT_O)

1247 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/usb_mid/%.c
1248 $(COMPILE.c) -o $$@ $<
1249 $(CTFCONVERT_O)

```

```

1251 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/usb_ia/%.c
1252 $(COMPILE.c) -o $$@ $<
1253 $(CTFCONVERT_O)

1255 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/usba/%.c
1256 $(COMPILE.c) -o $$@ $<
1257 $(CTFCONVERT_O)

1259 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/usba10/%.c
1260 $(COMPILE.c) -o $$@ $<
1261 $(CTFCONVERT_O)

1263 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/usb/hwa/hwahc/%.c
1264 $(COMPILE.c) -o $$@ $<
1265 $(CTFCONVERT_O)

1267 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/uwb/uwba/%.c
1268 $(COMPILE.c) -o $$@ $<
1269 $(CTFCONVERT_O)

1271 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/vuidmice/%.c
1272 $(COMPILE.c) -o $$@ $<
1273 $(CTFCONVERT_O)

1275 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/vnic/%.c
1276 $(COMPILE.c) -o $$@ $<
1277 $(CTFCONVERT_O)

1279 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/wpi/%.c
1280 $(COMPILE.c) -o $$@ $<
1281 $(CTFCONVERT_O)

1283 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/zyd/%.c
1284 $(COMPILE.c) -o $$@ $<
1285 $(CTFCONVERT_O)

1287 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/chxge/com/%.c
1288 $(COMPILE.c) -o $$@ $<
1289 $(CTFCONVERT_O)

1291 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/chxge/%.c
1292 $(COMPILE.c) -o $$@ $<
1293 $(CTFCONVERT_O)

1295 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/cxgbe/common/%.c
1296 $(COMPILE.c) -o $$@ $<
1297 $(CTFCONVERT_O)

1299 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/cxgbe/shared/%.c
1300 $(COMPILE.c) -o $$@ $<
1301 $(CTFCONVERT_O)

1303 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/cxgbe/firmware/%.c
1304 $(COMPILE.c) -o $$@ $<
1305 $(CTFCONVERT_O)

1307 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/cxgbe/t4nex/%.c
1308 $(COMPILE.c) -o $$@ $<
1309 $(CTFCONVERT_O)

1311 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/cxgbe/cxgbe/%.c
1312 $(COMPILE.c) -o $$@ $<
1313 $(CTFCONVERT_O)

1315 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ixgb/%.c

```



```

1316 $(COMPILE.c) -o $$@ $<
1317 $(CTFCONVERT_O)

1319 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/xge/drv/%.c
1320 $(COMPILE.c) -o $$@ $<
1321 $(CTFCONVERT_O)

1323 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/xge/hal/xgehal/%.c
1324 $(COMPILE.c) -o $$@ $<
1325 $(CTFCONVERT_O)

1327 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/e1000g/%.c
1328 $(COMPILE.c) -o $$@ $<
1329 $(CTFCONVERT_O)

1331 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/igb/%.c
1332 $(COMPILE.c) -o $$@ $<
1333 $(CTFCONVERT_O)

1335 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/iprb/%.c
1336 $(COMPILE.c) -o $$@ $<
1337 $(CTFCONVERT_O)

1339 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ixgbe/%.c
1340 $(COMPILE.c) -o $$@ $<
1341 $(CTFCONVERT_O)

1343 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ntxn/%.c
1344 $(COMPILE.c) -o $$@ $<
1345 $(CTFCONVERT_O)

1347 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/myril0ge/drv/%.c
1348 $(COMPILE.c) -o $$@ $<
1349 $(CTFCONVERT_O)

1351 $(OBJSDIR)/%.o: $(UTSBASE)/common/ipp/%.c
1352 $(COMPILE.c) -o $$@ $<
1353 $(CTFCONVERT_O)

1355 $(OBJSDIR)/%.o: $(UTSBASE)/common/ipp/ipgpc/%.c
1356 $(COMPILE.c) -o $$@ $<
1357 $(CTFCONVERT_O)

1359 $(OBJSDIR)/%.o: $(UTSBASE)/common/ipp/dlcosmk/%.c
1360 $(COMPILE.c) -o $$@ $<
1361 $(CTFCONVERT_O)

1363 $(OBJSDIR)/%.o: $(UTSBASE)/common/ipp/flowacct/%.c
1364 $(COMPILE.c) -o $$@ $<
1365 $(CTFCONVERT_O)

1367 $(OBJSDIR)/%.o: $(UTSBASE)/common/ipp/dscpmk/%.c
1368 $(COMPILE.c) -o $$@ $<
1369 $(CTFCONVERT_O)

1371 $(OBJSDIR)/%.o: $(UTSBASE)/common/ipp/meters/%.c
1372 $(COMPILE.c) -o $$@ $<
1373 $(CTFCONVERT_O)

1375 $(OBJSDIR)/%.o: $(UTSBASE)/common/kiconv/kiconv_emea/%.c
1376 $(COMPILE.c) -o $$@ $<
1377 $(CTFCONVERT_O)

1379 $(OBJSDIR)/%.o: $(UTSBASE)/common/kiconv/kiconv_ja/%.c
1380 $(COMPILE.c) -o $$@ $<
1381 $(CTFCONVERT_O)

```

```

1383 $(OBJSDIR)/%.o: $(UTSBASE)/common/kiconv/kiconv_ko/%.c
1384 $(COMPILE.c) -o $$@ $<
1385 $(CTFCONVERT_O)

1387 $(OBJSDIR)/%.o: $(UTSBASE)/common/kiconv/kiconv_sc/%.c
1388 $(COMPILE.c) -o $$@ $<
1389 $(CTFCONVERT_O)

1391 $(OBJSDIR)/%.o: $(UTSBASE)/common/kiconv/kiconv_tc/%.c
1392 $(COMPILE.c) -o $$@ $<
1393 $(CTFCONVERT_O)

1395 $(OBJSDIR)/%.o: $(UTSBASE)/common/kmdb/%.c
1396 $(COMPILE.c) -o $$@ $<
1397 $(CTFCONVERT_O)

1399 $(OBJSDIR)/%.o: $(UTSBASE)/common/ktli/%.c
1400 $(COMPILE.c) -o $$@ $<
1401 $(CTFCONVERT_O)

1403 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/scsi/adapters/iscsi/%.c
1404 $(COMPILE.c) -o $$@ $<
1405 $(CTFCONVERT_O)

1407 $(OBJSDIR)/%.o: $(COMMONBASE)/iscsi/%.c
1408 $(COMPILE.c) -o $$@ $<
1409 $(CTFCONVERT_O)

1411 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/kifconf/%.c
1412 $(COMPILE.c) -o $$@ $<
1413 $(CTFCONVERT_O)

1415 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/vr/%.c
1416 $(COMPILE.c) -o $$@ $<
1417 $(CTFCONVERT_O)

1419 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/yge/%.c
1420 $(COMPILE.c) -o $$@ $<
1421 $(CTFCONVERT_O)

1423 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/virtio/%.c
1424 $(COMPILE.c) -o $$@ $<
1425 $(CTFCONVERT_O)

1427 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/vioblk/%.c
1428 $(COMPILE.c) -o $$@ $<
1429 $(CTFCONVERT_O)

1431 #
1432 # krtld must refer to its own bzero/bcopy until the kernel is fully linked
1433 #
1434 $(OBJSDIR)/bootrd.o := CPPFLAGS += -DKOBJ_OVERRIDES
1435 $(OBJSDIR)/doreloc.o := CPPFLAGS += -DKOBJ_OVERRIDES
1436 $(OBJSDIR)/kobj.o := CPPFLAGS += -DKOBJ_OVERRIDES
1437 $(OBJSDIR)/kobj_boot.o := CPPFLAGS += -DKOBJ_OVERRIDES
1438 $(OBJSDIR)/kobj_bootflags.o := CPPFLAGS += -DKOBJ_OVERRIDES
1439 $(OBJSDIR)/kobj_convrelstr.o := CPPFLAGS += -DKOBJ_OVERRIDES
1440 $(OBJSDIR)/kobj_isa.o := CPPFLAGS += -DKOBJ_OVERRIDES
1441 $(OBJSDIR)/kobj_kdi.o := CPPFLAGS += -DKOBJ_OVERRIDES
1442 $(OBJSDIR)/kobj_lm.o := CPPFLAGS += -DKOBJ_OVERRIDES
1443 $(OBJSDIR)/kobj_reloc.o := CPPFLAGS += -DKOBJ_OVERRIDES
1444 $(OBJSDIR)/kobj_stubs.o := CPPFLAGS += -DKOBJ_OVERRIDES
1445 $(OBJSDIR)/kobj_subr.o := CPPFLAGS += -DKOBJ_OVERRIDES

1447 $(OBJSDIR)/%.o: $(UTSBASE)/common/krtld/%.c

```

```

1448 $(COMPILE.c) -o $@ $<
1449 $(CTFCONVERT_O)

1451 $(OBJS_DIR)/%.o: $(COMMONBASE)/list/%.c
1452 $(COMPILE.c) -o $@ $<
1453 $(CTFCONVERT_O)

1455 $(OBJS_DIR)/%.o: $(COMMONBASE)/lvm/%.c
1456 $(COMPILE.c) -o $@ $<
1457 $(CTFCONVERT_O)

1459 $(OBJS_DIR)/%.o: $(COMMONBASE)/lzma/%.c
1460 $(COMPILE.c) -o $@ $<
1461 $(CTFCONVERT_O)

1463 $(OBJS_DIR)/%.o: $(COMMONBASE)/crypto/md4/%.c
1464 $(COMPILE.c) -o $@ $<
1465 $(CTFCONVERT_O)

1467 $(OBJS_DIR)/%.o: $(COMMONBASE)/crypto/md5/%.c
1468 $(COMPILE.c) -o $@ $<
1469 $(CTFCONVERT_O)

1471 $(OBJS_DIR)/%.o: $(COMMONBASE)/net/dhcp/%.c
1472 $(COMPILE.c) -o $@ $<
1473 $(CTFCONVERT_O)

1475 $(OBJS_DIR)/%.o: $(COMMONBASE)/nvpair/%.c
1476 $(COMPILE.c) -o $@ $<
1477 $(CTFCONVERT_O)

1479 $(OBJS_DIR)/%.o: $(UTSBASE)/common/os/%.c
1480 $(COMPILE.c) -o $@ $<
1481 $(CTFCONVERT_O)

1483 $(OBJS_DIR)/%.o: $(UTSBASE)/common/pcmcia/cis/%.c
1484 $(COMPILE.c) -o $@ $<
1485 $(CTFCONVERT_O)

1487 $(OBJS_DIR)/%.o: $(UTSBASE)/common/pcmcia/cs/%.c
1488 $(COMPILE.c) -o $@ $<
1489 $(CTFCONVERT_O)

1491 $(OBJS_DIR)/%.o: $(UTSBASE)/common/pcmcia/nexus/%.c
1492 $(COMPILE.c) -o $@ $<
1493 $(CTFCONVERT_O)

1495 $(OBJS_DIR)/%.o: $(UTSBASE)/common/pcmcia/pcs/%.c
1496 $(COMPILE.c) -o $@ $<
1497 $(CTFCONVERT_O)

1499 $(OBJS_DIR)/%.o: $(UTSBASE)/common/rpc/%.c
1500 $(COMPILE.c) -o $@ $<
1501 $(CTFCONVERT_O)

1503 $(OBJS_DIR)/%.o: $(UTSBASE)/common/rpc/sec/%.c
1504 $(COMPILE.c) -o $@ $<
1505 $(CTFCONVERT_O)

1507 $(OBJS_DIR)/%.o: $(UTSBASE)/common/rpc/sec_gss/%.c
1508 $(COMPILE.c) -o $@ $<
1509 $(CTFCONVERT_O)

1511 $(OBJS_DIR)/%.o: $(COMMONBASE)/crypto/shal/%.c
1512 $(COMPILE.c) -o $@ $<
1513 $(CTFCONVERT_O)

```

```

1515 $(OBJS_DIR)/%.o: $(COMMONBASE)/crypto/sha2/%.c
1516 $(COMPILE.c) -o $@ $<
1517 $(CTFCONVERT_O)

1519 $(OBJS_DIR)/%.o: $(UTSBASE)/common/syscall/%.c
1520 $(COMPILE.c) -o $@ $<
1521 $(CTFCONVERT_O)

1523 $(OBJS_DIR)/%.o: $(UTSBASE)/common/tnf/%.c
1524 $(COMPILE.c) -o $@ $<
1525 $(CTFCONVERT_O)

1527 $(OBJS_DIR)/%.o: $(COMMONBASE)/tsol/%.c
1528 $(COMPILE.c) -o $@ $<
1529 $(CTFCONVERT_O)

1531 $(OBJS_DIR)/%.o: $(COMMONBASE)/util/%.c
1532 $(COMPILE.c) -o $@ $<
1533 $(CTFCONVERT_O)

1535 $(OBJS_DIR)/%.o: $(COMMONBASE)/unicode/%.c
1536 $(COMPILE.c) -o $@ $<
1537 $(CTFCONVERT_O)

1539 $(OBJS_DIR)/%.o: $(UTSBASE)/common/vm/%.c
1540 $(COMPILE.c) -o $@ $<
1541 $(CTFCONVERT_O)

1543 $(OBJS_DIR)/%.o: $(UTSBASE)/common/zmod/%.c
1544 $(COMPILE.c) -o $@ $<
1545 $(CTFCONVERT_O)

1547 $(OBJS_DIR)/zlib_obj.o: $(ZLIB_OBJS:%=$(OBJS_DIR)/%)
1548 $(LD) -r -Breduce -M$(UTSBASE)/common/zmod/mapfile -o $@ \
1549 $(ZLIB_OBJS:%=$(OBJS_DIR)/%)
1550 $(CTFMERGE) -t -f -L VERSION -o $@ $(ZLIB_OBJS:%=$(OBJS_DIR)/%)

1552 $(OBJS_DIR)/%.o: $(UTSBASE)/common/io/hxge/%.c
1553 $(COMPILE.c) -o $@ $<
1554 $(CTFCONVERT_O)

1556 $(OBJS_DIR)/%.o: $(UTSBASE)/common/io/tpm/%.c
1557 $(COMPILE.c) -o $@ $<
1558 $(CTFCONVERT_O)

1560 $(OBJS_DIR)/%.o: $(UTSBASE)/common/io/tpm/%.s
1561 $(COMPILE.s) -o $@ $<

1563 $(OBJS_DIR)/bz2%.o: $(COMMONBASE)/bzip2/%.c
1564 $(COMPILE.c) -o $@ -I$(COMMONBASE)/bzip2 $<
1565 $(CTFCONVERT_O)

1567 BZ2LINT = -erroff=%all -I$(UTSBASE)/common/bzip2

1569 $(LINTS_DIR)/bz2%.ln: $(COMMONBASE)/bzip2/%.c
1570 @($(LHEAD) $(LINT.c) -C $(LINTS_DIR)/`basename $@ .ln` $(BZ2LINT) $< $(

1572 #
1573 # SVM
1574 #

1576 MD_XDR_CSRC = $(UTSBASE)/common/io/lvm/md
1577 MD_XDR_XSRC = $(UTSBASE)/common/sys/lvm
1578 RPCGENFLAGS += -C -M -D_KERNEL -DSYSV

```



```

1708      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1710 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/disp/%.c
1711      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1713 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/dtrace/%.c
1714      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1716 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/exacct/%.c
1717      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1719 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/exec/aout/%.c
1720      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1722 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/exec/elf/%.c
1723      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1725 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/exec/intp/%.c
1726      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1728 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/exec/shbin/%.c
1729      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1731 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/exec/java/%.c
1732      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1734 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/%.c
1735      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1737 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/autofs/%.c
1738      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1740 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/cacheufs/%.c
1741      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1743 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/ctfs/%.c
1744      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1746 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/doorfs/%.c
1747      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1749 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/dcfis/%.c
1750      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1752 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/devfs/%.c
1753      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1755 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/dev/%.c
1756      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1758 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/fd/%.c
1759      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1761 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/fifofs/%.c
1762      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1764 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/hsfs/%.c
1765      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1767 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/lofs/%.c
1768      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1770 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/mntfs/%.c
1771      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1773 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/namefs/%.c

```

```

1774      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1776 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/smbsrv/%.c
1777      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1779 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/smbsrv/%.c
1780      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1782 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/nfs/%.c
1783      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1785 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/objfs/%.c
1786      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1788 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/pcfs/%.c
1789      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1791 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/portfs/%.c
1792      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1794 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/proc/%.c
1795      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1797 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/sharefs/%.c
1798      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1800 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/smbclnt/%.c
1801      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1803 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/smbclnt/net smb/%.c
1804      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1806 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/smbclnt/smbfs/%.c
1807      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1809 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/sockfs/%.c
1810      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1812 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/specfs/%.c
1813      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1815 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/swapfs/%.c
1816      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1818 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/tmpfs/%.c
1819      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1821 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/udfs/%.c
1822      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1824 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/ufs/%.c
1825      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1827 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/ufs_log/%.c
1828      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1830 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/vscan/%.c
1831      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1833 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/zfs/%.c
1834      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1836 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/zut/%.c
1837      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1839 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/xattr/%.c

```

```

1840      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1842 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/zfs/%.c
1843      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1845 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/gssapi/%.c
1846      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1848 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/gssapi/mechs/dummy/%.c
1849      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1851 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/%.c
1852      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1854 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/%.c
1855      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1857 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/des/%.c
1858      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1860 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/dk/%.c
1861      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1863 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/os/%.c
1864      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1866 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/arcfour/%.c
1867      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1869 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/enc_provider/%.c
1870      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1872 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/hash_provider/%.c
1873      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1875 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/keyhash_provider/%.c
1876      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1878 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/raw/%.c
1879      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1881 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/old/%.c
1882      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1884 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/krb5/krb/%.c
1885      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1887 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/krb5/os/%.c
1888      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1890 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/mech/%.c
1891      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1893 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/idmap/%.c
1894      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1896 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/%.c
1897      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1899 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/sockmods/%.c
1900      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1902 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/arp/%.c
1903      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1905 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/ip/%.c

```

```

1906      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1908 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/ipnet/%.c
1909      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1911 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/iptun/%.c
1912      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1914 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/ipf/%.c
1915      @$(LHEAD) $(LINT.c) $(IPFFLAGS) $< $(LTAIL))
1917 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/kssl/%.c
1918      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1920 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/net/patricia/%.c
1921      @$(LHEAD) $(LINT.c) $(IPFFLAGS) $< $(LTAIL))
1923 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/udp/%.c
1924      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1926 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/sctp/%.c
1927      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1929 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/tcp/%.c
1930      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1932 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/ilb/%.c
1933      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1935 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/nca/%.c
1936      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1938 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/dlpistub/%.c
1939      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1941 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/%.c
1942      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1944 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/1394/%.c
1945      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1947 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/1394/adapters/%.c
1948      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1950 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/1394/targets/av1394/%.c
1951      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1953 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/1394/targets/dcam1394/%.c
1954      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1956 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/1394/targets/scsal394/%.c
1957      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1959 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sbp2/%.c
1960      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1962 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/aac/%.c
1963      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1965 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/afe/%.c
1966      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1968 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/atge/%.c
1969      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1971 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/arn/%.c

```

```

1972      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1974 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ath/%.c
1975      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1977 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/atu/%.c
1978      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1980 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/impl/%.c
1981      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1983 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/ac97/%.c
1984      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1986 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audio1575/%.c
1987      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1989 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audio810/%.c
1990      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1992 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiocmi/%.c
1993      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1995 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiocmihd/%.c
1996      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1998 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audioens/%.c
1999      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2001 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audioemu10k/%.c
2002      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2004 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiohd/%.c
2005      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2007 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audioixp/%.c
2008      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2010 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiols/%.c
2011      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2013 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiopci/%.c
2014      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2016 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiopl6x/%.c
2017      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2019 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiosolo/%.c
2020      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2022 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiots/%.c
2023      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2025 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiovia823x/%.c
2026      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2028 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiovia97/%.c
2029      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2031 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/bfe/%.c
2032      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2034 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/bpf/%.c
2035      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2037 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/bge/%.c

```

```

2038      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2040 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/blkdev/%.c
2041      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2043 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/cardbus/%.c
2044      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2046 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/lu/stmf_sbd/%.c
2047      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2049 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/port/fct/%.c
2050      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2052 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/port/qlt/%.c
2053      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2055 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/port/srpt/%.c
2056      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2058 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/iscsit/%.c
2059      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2061 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/port/fcoet/%.c
2062      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2064 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/port/iscsit/%.c
2065      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2067 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/port/pppt/%.c
2068      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2070 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/stmf/%.c
2071      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2073 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/dld/%.c
2074      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2076 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/dls/%.c
2077      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2079 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/dmfe/%.c
2080      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2082 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/drm/%.c
2083      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2085 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/efe/%.c
2086      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2088 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/elxl/%.c
2089      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2091 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fcoe/%.c
2092      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2094 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/hme/%.c
2095      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2097 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pciex/%.c
2098      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2100 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/hotplug/hpcsvc/%.c
2101      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2103 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pciex/hotplug/%.c

```

```

2104      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2106 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/hotplug/pcihp/%.c
2107      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2109 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/rds/%.c
2110      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2112 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/rds3/%.c
2113      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2115 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/iser/%.c
2116      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2118 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/ibd/%.c
2119      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2121 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/eoib/%.c
2122      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2124 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/of/sol_ofs/%.c
2125      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2127 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/of/sol_ucma/%.c
2128      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2130 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/of/sol_umad/%.c
2131      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2133 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/of/sol_uverbs/%.
2134      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2136 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/sdp/%.c
2137      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2139 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/mgt/ibcm/%.c
2140      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2142 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/mgt/ibdm/%.c
2143      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2145 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/mgt/ibdma/%.c
2146      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2148 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/mgt/ibmf/%.c
2149      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2151 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/ibnex/%.c
2152      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2154 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/ibt1/%.c
2155      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2157 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/adapters/tavor/%.c
2158      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2160 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/adapters/hermon/%.c
2161      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2163 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/dapl1/%.c
2164      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2166 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/iscsi/%.c
2167      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2169 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/idm/%.c

```

```

2170      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2172 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ipw/%.c
2173      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2175 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/iwh/%.c
2176      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2178 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/iwi/%.c
2179      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2181 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/iwk/%.c
2182      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2184 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/iwp/%.c
2185      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2187 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/kb8042/%.c
2188      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2190 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/kbtrans/%.c
2191      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2193 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ksocket/%.c
2194      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2196 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/aggr/%.c
2197      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2199 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lp/%.c
2200      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2202 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/hotspares/%.c
2203      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2205 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/md/%.c
2206      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2208 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/mirror/%.c
2209      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2211 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/raid/%.c
2212      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2214 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/softpart/%.c
2215      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2217 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/striped/%.c
2218      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2220 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/notify/%.c
2221      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2223 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/trans/%.c
2224      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2226 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mac/%.c
2227      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2229 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mac/plugins/%.c
2230      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2232 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mega_sas/%.c
2233      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2235 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mii/%.c

```

```

2236      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2238 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mr_sas/%.c
2239      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2241 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/mpt_sas/%.c
2242      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2244 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mxfe/%.c
2245      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2247 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mwl/%.c
2248      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2250 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mwl/mwl_fw/%.c
2251      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2253 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/net80211/%.c
2254      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2256 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/nge/%.c
2257      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2259 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/nxge/%.c
2260      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2262 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/nxge/%.s
2263      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2265 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/nxge/npi/%.c
2266      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2268 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pci-ide/%.c
2269      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2271 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pcmcia/%.c
2272      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2274 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pcan/%.c
2275      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2277 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pcn/%.c
2278      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2280 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pcwl/%.c
2281      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2283 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ppp/sppp/%.c
2284      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2286 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ppp/spppasyn/%.c
2287      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2289 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ppp/spptun/%.c
2290      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2292 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ral/%.c
2293      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2295 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rge/%.c
2296      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2298 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rtls/%.c
2299      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2301 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rsm/%.c

```

```

2302      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2304 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rtw/%.c
2305      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2307 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rum/%.c
2308      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2310 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rwd/%.c
2311      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2313 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rwn/%.c
2314      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2316 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sata/adapters/ahci/%.c
2317      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2319 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sata/adapters/nv_sata/%.c
2320      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2322 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sata/adapters/si3124/%.c
2323      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2325 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sata/impl/%.c
2326      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2328 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/%.c
2329      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2331 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/blk2scsa/%.c
2332      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2334 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/pmcs/%.c
2335      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2337 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/%.c
2338      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2340 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/fop
2341      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2343 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/ulp/%.c
2344      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2346 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/impl/%.c
2347      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2349 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/fca/qlc/%.c
2350      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2352 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/fca/qlge/%.c
2353      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2355 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/fca/emlxs/%.c
2356      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2358 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/fca/oce/%.c
2359      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2361 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/fca/fcoei/%.c
2362      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2364 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/conf/%.c
2365      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2367 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/impl/%.c

```



```

2368      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2370 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/targets/%.c
2371      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2373 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sdcard/adapters/sdhost/%.c
2374      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2376 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sdcard/impl/%.c
2377      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2379 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sdcard/targets/sdcard/%.c
2380      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2382 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sfe/%.c
2383      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2385 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/simnet/%.c
2386      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2388 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/softmac/%.c
2389      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2391 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/uath/%.c
2392      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2394 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/uath/uath_fw/%.c
2395      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2397 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ural/%.c
2398      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2400 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/urtw/%.c
2401      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2403 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/audio/usb_ac/%.
2404      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2406 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/audio/usb_as/%.
2407      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2409 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/audio/usb_ah/%.
2410      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2412 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbskel/%.c
2413      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2415 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/video/usbvc/%.c
2416      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2418 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/hwarc/%.c
2419      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2421 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/hid/%.c
2422      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2424 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/hidparser/%.c
2425      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2427 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbkbm/%.c
2428      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2430 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbms/%.c
2431      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2433 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbinp/usbwcm

```

```

2434      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2436 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/ugen/%.c
2437      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2439 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/printer/%.c
2440      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2442 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbser/%.c
2443      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2445 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbser/usbsacm/
2446      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2448 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbser/usbftdi/
2449      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2451 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbser/usbser_k
2452      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2454 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbser/usbsprl/
2455      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2457 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/wusb_df/%.c
2458      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2460 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/hwal480_fw/%.c
2461      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2463 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/wusb_ca/%.c
2464      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2466 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbecm/%.c
2467      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2469 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/hcd/openhci/%.c
2470      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2472 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/hcd/ehci/%.c
2473      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2475 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/hcd/uhci/%.c
2476      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2478 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/hubd/%.c
2479      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2481 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/scsa2usb/%.c
2482      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2484 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/usb_mid/%.c
2485      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2487 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/usb_ia/%.c
2488      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2490 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/usba/%.c
2491      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2493 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/usba10/%.c
2494      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2496 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/uwb/uwba/%.c
2497      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2499 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/hwa/hwahc/%.c

```

```

2500      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2502 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/vuidmice/%.c
2503      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2505 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/vnic/%.c
2506      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2508 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/wpi/%.c
2509      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2511 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/zyd/%.c
2512      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2514 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/chxge/com/%.c
2515      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2517 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/chxge/%.c
2518      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2520 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/cxgbe/common/%.c
2521      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2523 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/cxgbe/shared/%.c
2524      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2526 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/cxgbe/firmware/%.c
2527      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2529 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/cxgbe/t4nex/%.c
2530      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2532 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/cxgbe/cxgbe/%.c
2533      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2535 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ixgb/%.c
2536      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2538 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/xge/drv/%.c
2539      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2541 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/xge/hal/xgehal/%.c
2542      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2544 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/e1000g/%.c
2545      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2547 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/igb/%.c
2548      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2550 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/iprb/%.c
2551      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2553 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ixgbe/%.c
2554      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2556 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ntxn/%.c
2557      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2559 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/myril0ge/drv/%.c
2560      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2562 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ipp/%.c
2563      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2565 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ipp/ipgpc/%.c

```

```

2566      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2568 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ipp/dlcosmk/%.c
2569      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2571 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ipp/flowacct/%.c
2572      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2574 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ipp/dscpmk/%.c
2575      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2577 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ipp/meters/%.c
2578      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2580 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/kiconv/kiconv_emea/%.c
2581      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2583 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/kiconv/kiconv_ja/%.c
2584      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2586 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/kiconv/kiconv_ko/%.c
2587      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2589 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/kiconv/kiconv_sc/%.c
2590      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2592 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/kiconv/kiconv_tc/%.c
2593      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2595 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/kmdb/%.c
2596      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2598 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/krtld/%.c
2599      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2601 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ktli/%.c
2602      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2604 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/list/%.c
2605      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2607 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/lvm/%.c
2608      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2610 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/lzma/%.c
2611      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2613 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/md4/%.c
2614      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2616 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/md5/%.c
2617      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2619 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/net/dhcp/%.c
2620      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2622 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/nvpair/%.c
2623      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2625 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/os/%.c
2626      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2628 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/rpc/%.c
2629      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2631 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/pcmcia/cs/%.c

```

```

2632      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2634 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/pcmcia/cis/%.c
2635      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2637 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/pcmcia/nexus/%.c
2638      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2640 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/pcmcia/pcs/%.c
2641      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2643 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/rpc/%.c
2644      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2646 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/rpc/sec/%.c
2647      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2649 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/rpc/sec_gss/%.c
2650      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2652 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/shal/%.c
2653      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2655 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/sha2/%.c
2656      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2658 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/syscall/%.c
2659      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2661 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/tnf/%.c
2662      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2664 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/tsol/%.c
2665      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2667 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/util/%.c
2668      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2670 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/unicode/%.c
2671      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2673 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/vm/%.c
2674      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2676 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/iscsi/%.c
2677      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2679 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/iscsi/%.c
2680      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2682 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/kifconf/%.c
2683      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2685 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/virtio/%.c
2686      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2688 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/vioblk/%.c
2689      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2691 ZMODLINTFLAGS = -erroff=E_CONSTANT_CONDITION
2693 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/zmod/%.c
2694      @$(LHEAD) $(LINT.c) $(ZMODLINTFLAGS) $< $(LTAIL))
2696 $(LINTS_DIR)/zlib_obj.ln: $(ZLIB_OBJS:%.o=$(LINTS_DIR)/%.ln) \
2697      $(UTSBASE)/common/zmod/zlib_lint.c

```

```

2698      @$(LHEAD) $(LINT.c) -C $(LINTS_DIR)/zlib_obj \
2699      $(UTSBASE)/common/zmod/zlib_lint.c $(LTAIL))
2701 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/hxge/%.c
2702      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2704 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/tpm/%.c
2705      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2707 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/tpm/%.s
2708      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2710 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/vr/%.c
2711      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2713 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/yge/%.c
2714      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2716 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/fsreparse/%.c
2717      @$(LHEAD) $(LINT.c) $< $(LTAIL))

```

```

*****
22710 Sun May 26 18:24:04 2013
new/usr/src/uts/common/sys/Makefile
XXX Remove awk(1)
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
23 #

25 include $(SRC)/uts/Makefile.uts

27 FILEMODE=644

29 #
30 # Note that the following headers are present in the kernel but
31 # neither installed or shipped as part of the product:
32 # cpuid_drv.h: Private interface for cpuid consumers
33 # unix_bb_info.h: Private interface to kcov
34 #

36 i386_HDRS= \
37 agp/agpamd64gart_io.h \
38 agp/agpdefs.h \
39 agp/agpgart_impl.h \
40 agp/agpmaster_io.h \
41 agp/agptarget_io.h \
42 agpgart.h \
43 asy.h \
44 fd_debug.h \
45 fdc.h \
46 fdmedia.h \
47 mouse.h \
48 ucode.h

50 sparc_HDRS= \
51 mouse.h \
52 scsi/targets/ssddef.h \
53 $(MDESCHDRS)

55 # Generated headers
56 GENHDRS= \
57 priv_const.h \
58 priv_names.h \
59 usb/usbdevs.h

61 CHKHDRS= \

```

```

62 acpi_drv.h \
63 acct.h \
64 acctctl.h \
65 acl.h \
66 acl_impl.h \
67 aggr.h \
68 aggr_impl.h \
69 aio.h \
70 aio_impl.h \
71 aio_req.h \
72 aiocb.h \
73 ascii.h \
74 asynch.h \
75 atomic.h \
76 attr.h \
77 audio.h \
78 audioio.h \
79 autoconf.h \
80 auxv.h \
81 auxv_386.h \
82 auxv_SPARC.h \
83 avl.h \
84 avl_impl.h \
85 bitmap.h \
86 bitset.h \
87 bl.h \
88 blkdev.h \
89 bofi.h \
90 bofi_impl.h \
91 bpp_io.h \
92 bootstat.h \
93 brand.h \
94 buf.h \
95 bufmod.h \
96 bustypes.h \
97 byteorder.h \
98 callb.h \
99 callo.h \
100 cap_util.h \
101 cpucaps.h \
102 cpucaps_impl.h \
103 ccompile.h \
104 cdio.h \
105 cladm.h \
106 class.h \
107 clconf.h \
108 clock_impl.h \
109 cmlb.h \
110 cmn_err.h \
111 compress.h \
112 condvar.h \
113 condvar_impl.h \
114 conf.h \
115 consdev.h \
116 console.h \
117 consplat.h \
118 vt.h \
119 vtdaemon.h \
120 kd.h \
121 contract.h \
122 contract_impl.h \
123 copyops.h \
124 core.h \
125 corectl.h \
126 cpc_impl.h \
127 cpc_pcbe.h \

```

```

128     cpr.h                \|
129     cpupart.h           \|
130     cpuvar.h            \|
131     crc32.h             \|
132     cred.h              \|
133     cred_impl.h        \|
134     crtctl.h            \|
135     cryptmod.h          \|
136     csiiioctl.h         \|
137     ctf.h               \|
138     ctfs.h              \|
139     ctfs_impl.h        \|
140     ctf_api.h           \|
141     ctype.h             \|
142     cyclic.h            \|
143     cyclic_impl.h      \|
144     dacf.h              \|
145     dacf_impl.h        \|
146     damap.h             \|
147     damap_impl.h       \|
148     dc_ki.h             \|
149     ddi.h               \|
150     ddifm.h             \|
151     ddifm_impl.h       \|
152     ddi_hp.h            \|
153     ddi_hp_impl.h      \|
154     ddi_intr.h          \|
155     ddi_intr_impl.h    \|
156     ddi_impldefs.h     \|
157     ddi_implfuncs.h    \|
158     ddi_obsolete.h     \|
159     ddi_timer.h         \|
160     ddidevmap.h         \|
161     ddidmareq.h         \|
162     ddimapreq.h         \|
163     ddipropdefs.h     \|
164     ddiypes.h           \|
165     debug.h             \|
166     des.h               \|
167     devctl.h            \|
168     devcache.h          \|
169     devcache_impl.h    \|
170     devfm.h             \|
171     devid_cache.h       \|
172     devinfo_impl.h     \|
173     devops.h            \|
174     devpolicy.h         \|
175     devpoll.h           \|
176     dirent.h            \|
177     disp.h              \|
178     dkbad.h             \|
179     dkio.h              \|
180     dklabel.h           \|
181     dl.h                \|
182     dlpi.h              \|
183     dld.h               \|
184     dld_impl.h          \|
185     dld_ioc.h           \|
186     dls.h               \|
187     dls_mgmt.h          \|
188     dls_impl.h          \|
189     dma_i8237A.h        \|
190     dnlc.h              \|
191     door.h              \|
192     door_data.h         \|
193     door_impl.h         \|

```

```

194     dtrace.h            \|
195     dtrace_impl.h       \|
196     dumpadm.h           \|
197     dumphdr.h           \|
198     ecppsys.h           \|
199     ecppio.h            \|
200     ecppreg.h           \|
201     ecppvar.h           \|
202     efi_partition.h     \|
203     elf.h               \|
204     elf_386.h           \|
205     elf_SPARC.h         \|
206     elf_notes.h         \|
207     elf_amd64.h         \|
208     elftypes.h          \|
209     emul64.h            \|
210     emul64cmd.h         \|
211     emul64var.h         \|
212     epm.h                \|
213     errno.h             \|
214     errorq.h            \|
215     errorq_impl.h       \|
216     esunddi.h           \|
217     ethernet.h          \|
218     euc.h                \|
219     eucliocntl.h        \|
220     exacctl.h           \|
221     exacctl_catalog.h   \|
222     exacctl_impl.h      \|
223     exec.h               \|
224     exechnr.h           \|
225     extdirent.h         \|
226     fault.h             \|
227     fasttrap.h          \|
228     fasttrap_impl.h     \|
229     fbio.h              \|
230     fbuf.h              \|
231     fcntl.h             \|
232     fct.h               \|
233     fct_defines.h       \|
234     fctio.h             \|
235     fdbuffer.h          \|
236     fdio.h              \|
237     feature_tests.h     \|
238     fem.h                \|
239     file.h              \|
240     filio.h             \|
241     flock.h             \|
242     flock_impl.h        \|
243     fork.h              \|
244     fss.h               \|
245     fssprioctl.h        \|
246     fsid.h              \|
247     fssnap.h            \|
248     fssnap_if.h         \|
249     fstyp.h             \|
250     ftrace.h            \|
251     fx.h                \|
252     fxprioctl.h         \|
253     gfs.h               \|
254     gid.h               \|
255     gldpriv.h           \|
256     group.h             \|
257     hdio.h              \|
258     hook.h              \|
259     hook_event.h        \|

```

```

260     hook_impl.h      \|
261     hwconf.h         \|
262     ia.h              \|
263     iapriocntl.h     \|
264     ibpart.h         \|
265     id32.h           \|
266     idmap.h          \|
267     ieeeep.h         \|
268     id_space.h       \|
269     instance.h       \|
270     int_const.h      \|
271     int_fmtio.h      \|
272     int_limits.h     \|
273     int_types.h      \|
274     inttypes.h       \|
275     iocom.h          \|
276     ioctl.h          \|
277     ipc.h            \|
278     ipc_impl.h       \|
279     ipc_rctl.h       \|
280     ipmi.h           \|
281     isa_defs.h       \|
282     iscsi_authclient.h \|
283     iscsi_authclientglue.h \|
284     iscsi_protocol.h \|
285     jioctl.h         \|
286     kbd.h            \|
287     kbdreg.h         \|
288     kbio.h           \|
289     kcpic.h          \|
290     kdi.h            \|
291     kdi_impl.h       \|
292     kiconv.h         \|
293     kiconv_big5_utf8.h \|
294     kiconv_ck_common.h \|
295     kiconv_cp950hkscs_utf8.h \|
296     kiconv_emea1.h   \|
297     kiconv_emea2.h   \|
298     kiconv_euckr_utf8.h \|
299     kiconv_euctw_utf8.h \|
300     kiconv_gb18030_utf8.h \|
301     kiconv_gb2312_utf8.h \|
302     kiconv_hkscs_utf8.h \|
303     kiconv_ja.h      \|
304     kiconv_ja_jis_to_unicode.h \|
305     kiconv_ja_unicode_to_jis.h \|
306     kiconv_ko.h      \|
307     kiconv_latin1.h \|
308     kiconv_sc.h      \|
309     kiconv_tc.h      \|
310     kiconv_uhc_utf8.h \|
311     kiconv_utf8_big5.h \|
312     kiconv_utf8_cp950hkscs.h \|
313     kiconv_utf8_euckr.h \|
314     kiconv_utf8_euctw.h \|
315     kiconv_utf8_gb18030.h \|
316     kiconv_utf8_gb2312.h \|
317     kiconv_utf8_hkscs.h \|
318     kiconv_utf8_uhc.h \|
319     kidmap.h         \|
320     klpd.h           \|
321     klwp.h           \|
322     kmdb.h           \|
323     kmem.h           \|
324     kmem_impl.h     \|
325     kobj.h           \|

```

```

326     kobj_impl.h     \|
327     ksocket.h       \|
328     kstat.h         \|
329     kstr.h          \|
330     ksyms.h         \|
331     ksynch.h        \|
332     ldterm.h        \|
333     lgrp.h          \|
334     lgrp_user.h     \|
335     libc_kernel.h   \|
336     link.h          \|
337     list.h          \|
338     list_impl.h     \|
339     llc1.h          \|
340     loadavg.h       \|
341     lock.h          \|
342     lockfs.h        \|
343     lockstat.h     \|
344     lofi.h          \|
345     log.h           \|
346     logindmux.h     \|
347     logindmux_impl.h \|
348     lwp.h           \|
349     lwp_timer_impl.h \|
350     lwp_upimutex_impl.h \|
351     lpif.h          \|
352     mac.h           \|
353     mac_client.h    \|
354     mac_client_impl.h \|
355     mac_ether.h     \|
356     mac_flow.h      \|
357     mac_flow_impl.h \|
358     mac_impl.h      \|
359     mac_provider.h  \|
360     mac_soft_ring.h \|
361     mac_stat.h      \|
362     machelf.h       \|
363     map.h           \|
364     md4.h           \|
365     md5.h           \|
366     md5_consts.h   \|
367     mdi_impldefs.h \|
368     mem.h           \|
369     mem_config.h    \|
370     memlist.h       \|
371     mkdev.h         \|
372     mhd.h           \|
373     mii.h           \|
374     miiregs.h       \|
375     mixer.h         \|
376     mman.h          \|
377     mmapobj.h       \|
378     mntent.h        \|
379     mntio.h         \|
380     mnttab.h        \|
381     modctl.h        \|
382     mode.h          \|
383     model.h         \|
384     modhash.h       \|
385     modhash_impl.h \|
386     mount.h         \|
387     mouse.h         \|
388     msacct.h        \|
389     msg.h           \|
390     msg_impl.h      \|
391     msio.h          \|

```

```

392     msreg.h           \|
393     mtio.h            \|
394     multidata.h      \|
395     multidata_impl.h \|
396     mutex.h          \|
397     nbmlock.h        \|
398     ndifm.h          \|
399     ndi_impldefs.h   \|
400     net80211.h       \|
401     net80211_crypto.h \|
402     net80211_ht.h   \|
403     net80211_proto.h \|
404     netconfig.h      \|
405     neti.h           \|
406     netstack.h       \|
407     nexusdefs.h     \|
408     note.h           \|
409     nvpair.h         \|
410     nvpair_impl.h   \|
411     objfs.h          \|
412     objfs_impl.h    \|
413     ontrap.h         \|
414     open.h           \|
415     openpromio.h    \|
416     panic.h         \|
417     param.h         \|
418     pathconf.h      \|
419     pathname.h      \|
420     pattr.h         \|
421     queue.h         \|
422     serializer.h    \|
423     pbio.h          \|
424     pccard.h        \|
425     pci.h           \|
426     pcie.h          \|
427     pci_impl.h      \|
428     pci_tools.h     \|
429     pcmcia.h        \|
430     pctype.h        \|
431     pfmod.h         \|
432     pg.h            \|
433     pghw.h          \|
434     physmem.h       \|
435     pkp_hash.h      \|
436     pm.h            \|
437     policy.h        \|
438     poll.h          \|
439     poll_impl.h     \|
440     pool.h          \|
441     pool_impl.h     \|
442     pool_pset.h     \|
443     port.h          \|
444     port_impl.h     \|
445     port_kernel.h   \|
446     portif.h        \|
447     ppmio.h         \|
448     pppt_ic_if.h    \|
449     pppt_ioctl.h    \|
450     priocntl.h      \|
451     priv.h          \|
452     priv_impl.h     \|
453     prnio.h         \|
454     proc.h          \|
455     processor.h     \|
456     procfb.h        \|
457     procset.h       \|

```

```

458     project.h       \|
459     protosw.h       \|
460     prsystem.h      \|
461     pset.h          \|
462     pshot.h         \|
463     ptem.h          \|
464     ptms.h          \|
465     ptyvar.h        \|
466     raidioctl.h     \|
467     ramdisk.h       \|
468     random.h        \|
469     rctl.h          \|
470     rctl_impl.h     \|
471     rds.h           \|
472     reboot.h        \|
473     refstr.h        \|
474     refstr_impl.h   \|
475     resource.h      \|
476     rliocntl.h     \|
477     rt.h            \|
478     rtpricntl.h     \|
479     rwlock.h        \|
480     rwlock_impl.h  \|
481     rwstlock.h     \|
482     sad.h           \|
483     schedctl.h      \|
484     sdt.h           \|
485     select.h        \|
486     sem.h           \|
487     sem_impl.h     \|
488     sema_impl.h     \|
489     semaphore.h     \|
490     sendfile.h      \|
491     ser_sync.h      \|
492     session.h       \|
493     shal.h          \|
494     shal_consts.h   \|
495     sha2.h          \|
496     sha2_consts.h   \|
497     share.h         \|
498     shm.h           \|
499     shm_impl.h      \|
500     sid.h           \|
501     siginfo.h       \|
502     signal.h        \|
503     sleepq.h        \|
504     sbios.h         \|
505     sbios_impl.h    \|
506     sobject.h       \|
507     socket.h        \|
508     socket_impl.h   \|
509     socket_proto.h  \|
510     socketvar.h     \|
511     sockfilter.h    \|
512     sockio.h        \|
513     soundcard.h     \|
514     squeue.h        \|
515     squeue_impl.h  \|
516     srn.h           \|
517     sservice.h      \|
518     stat.h          \|
519     statfs.h        \|
520     statvfs.h       \|
521     stdbool.h       \|
522     stdint.h        \|
523     stermio.h       \|

```

```

524     stmf.h                \|
525     stmf_defines.h       \|
526     stmf_ioctl.h        \|
527     stmf_sbd_ioctl.h    \|
528     stream.h            \|
529     strft.h             \|
530     strlog.h            \|
531     strmddep.h          \|
532     stropts.h           \|
533     strredir.h          \|
534     strstat.h           \|
535     strsubr.h           \|
536     strsun.h            \|
537     strtty.h            \|
538     sunddi.h            \|
539     sunldi.h            \|
540     sunldi_impl.h       \|
541     sunmdi.h            \|
542     sunndi.h            \|
543     sunos_dhcp_class.h  \|
544     sunpm.h             \|
545     suntpi.h            \|
546     suntty.h            \|
547     swap.h              \|
548     synch.h             \|
549     sysdc.h             \|
550     sysdc_impl.h        \|
551     syscall.h           \|
552     sysconf.h           \|
553     sysconfig.h         \|
554     sysevent.h          \|
555     sysevent_impl.h    \|
556     sysinfo.h           \|
557     syslog.h            \|
558     sysmacros.h         \|
559     sysmsg_impl.h       \|
560     systeminfo.h        \|
561     system.h            \|
562     task.h              \|
563     taskq.h             \|
564     taskq_impl.h        \|
565     t_kuser.h           \|
566     t_lock.h            \|
567     telioc1.h           \|
568     termio.h            \|
569     termios.h           \|
570     termiox.h           \|
571     thread.h            \|
572     ticlts.h            \|
573     ticots.h            \|
574     ticotsord.h         \|
575     tihdr.h             \|
576     time.h              \|
577     time_impl.h         \|
578     time_std_impl.h     \|
579     timeb.h             \|
580     timer.h             \|
581     times.h             \|
582     timex.h             \|
583     timod.h             \|
584     tirdwr.h            \|
585     tiuser.h            \|
586     tl.h                \|
587     tnf.h               \|
588     tnf_com.h           \|
589     tnf_probe.h         \|

```

```

590     tnf_writer.h        \|
591     todio.h             \|
592     tpicommon.h         \|
593     ts.h                \|
594     tspriocntl.h        \|
595     ttcompat.h          \|
596     ttold.h             \|
597     tty.h               \|
598     ttychars.h          \|
599     ttydev.h            \|
600     tuneable.h          \|
601     turnstile.h         \|
602     types.h             \|
603     types32.h           \|
604     tzfile.h            \|
605     u8_textprep.h       \|
606     u8_textprep_data.h \|
607     uadmin.h            \|
608     ucred.h             \|
609     uio.h               \|
610     ulimit.h            \|
611     un.h                \|
612     unistd.h            \|
613     user.h              \|
614     ustat.h             \|
615     utime.h             \|
616     utsname.h           \|
617     utssys.h            \|
618     uuid.h              \|
619     va_impl.h           \|
620     va_list.h           \|
621     var.h                \|
622     varargs.h           \|
623     vfs.h                \|
624     vfs_opreg.h         \|
625     vfstab.h            \|
626     vgareg.h            \|
627     videodev2.h         \|
628     visual_io.h         \|
629     vlan.h              \|
630     vm.h                \|
631     vm_usage.h          \|
632     vmem.h              \|
633     vmem_impl.h         \|
634     vmsystem.h          \|
635     vnic.h              \|
636     vnic_impl.h         \|
637     vnode.h             \|
638     vscan.h             \|
639     vtoc.h              \|
640     vtrace.h            \|
641     vuid_event.h        \|
642     vuid_wheel.h        \|
643     vuid_queue.h        \|
644     vuid_state.h        \|
645     vuid_store.h        \|
646     wait.h              \|
647     waitq.h             \|
648     wanboot_impl.h     \|
649     watchpoint.h        \|
650     winlockio.h         \|
651     zcons.h             \|
652     zone.h              \|
653     xti_inet.h          \|
654     xti_osi.h           \|
655     xti_xtiopt.h        \|

```



```

656      zmod.h
658 HDRS=
659      $(GENHDRS)
660      $(CHKHDRS)
662 AUDIOHDRS=
663      ac97.h
664      audio_common.h
665      audio_driver.h
666      audio_oss.h
667      g711.h
669 AVHDRS=
670      iec61883.h
672 BSCHDRS=
673      bscbus.h
674      bscv_impl.h
675      lom_ebuscodes.h
676      lom_io.h
677      lom_priv.h
678      lombus.h
680 MDESCHDRS=
681      mdesc.h
682      mdesc_impl.h
684 CPUDRVHDRS=
685      cpudrv.h
687 CRYPTOHDRS=
688      elfsign.h
689      ioctl.h
690      ioctladmin.h
691      common.h
692      impl.h
693      spi.h
694      api.h
695      ops_impl.h
696      sched_impl.h
698 DCAMHDRS=
699      dcaml394_io.h
701 IBHDRS=
702      ib_types.h
703      ib_pkt_hdrs.h
705 IBTLHDRS=
706      ibtl_types.h
707      ibtl_status.h
708      ibti.h
709      ibti_cm.h
710      ibci.h
711      ibti_common.h
712      ibvti.h
713      ibtl_ci_types.h
715 IBTLIMPLHDRS=
716      ibtl_util.h
718 IBNEXHDRS=
719      ibnex_devctl.h
721 IBMFHDRS=

```

```

722      ibmf.h
723      ibmf_msg.h
724      ibmf_saa.h
725      ibmf_utils.h
727 IBMGTHDRS=
728      ib_dm_attr.h
729      ib_mad.h
730      sm_attr.h
731      sa_recs.h
733 IBDHDRS=
734      ibd.h
736 OFHDRS=
737      ofa_solaris.h
738      ofed_kernel.h
740 RDMAHDRS=
741      ib_addr.h
742      ib_user_mad.h
743      ib_user_sa.h
744      ib_user_verbs.h
745      ib_verbs.h
746      rdma_cm.h
747      rdma_user_cm.h
749 SOL_UVERBSHDRS=
750      sol_uverbs.h
751      sol_uverbs2ucma.h
752      sol_uverbs_comp.h
753      sol_uverbs_hca.h
754      sol_uverbs_gp.h
755      sol_uverbs_event.h
757 SOL_UMADHDRS=
758      sol_umad.h
760 SOL_UCMAHDRS=
761      sol_ucma.h
762      sol_rdma_user_cm.h
764 SOL_OFSHDRS=
765      sol_cma.h
766      sol_ib_cma.h
767      sol_ofs_common.h
768      sol_kverb_impl.h
770 TAVORHDRS=
771      tavor_ioctl.h
773 HERMONHDRS=
774      hermon_ioctl.h
776 MLNXHDRS=
777      mlnx_umap.h
779 IDMHDRS=
780      idm.h
781      idm_impl.h
782      idm_so.h
783      idm_text.h
784      idm_transport.h
785      idm_conn_sm.h
787 ISCSITHDRS=

```

```

788     radius_packet.h  \
789     radius_protocol.h \
790     chap.h           \
791     isns_protocol.h  \
792     iscsi_if.h       \
793     iscsit_common.h  \

795 ISOHDRS= \
796     signal_iso.h \

798 DERIVED_LVMHDRS= \
799     md_mdiox.h    \
800     md_basic.h    \
801     mdmed.h       \
802     md_mhdx.h     \
803     mdmn_commd.h \

805 LVMHDRS= \
806     md_convert.h  \
807     md_crc.h       \
808     md_hotspares.h \
809     md_mddb.h      \
810     md_mirror.h   \
811     md_mirror_shared.h \
812     md_names.h    \
813     md_notify.h   \
814     md_raid.h     \
815     md_rename.h   \
816     md_sp.h       \
817     md_stripe.h   \
818     md_trans.h    \
819     mdio.h        \
820     mdvar.h       \

822 ALL_LVMHDRS= \
823     $(LVMHDRS) \
824     $(DERIVED_LVMHDRS) \

826 FMHDRS= \
827     protocol.h   \
828     util.h       \

830 FMFSHDRS= \
831     zfs.h        \

833 FMIOHDRS= \
834     ddi.h        \
835     disk.h       \
836     pci.h        \
837     scsi.h       \
838     sun4upci.h  \
839     opl_mc_fm.h  \

841 FSHDRS= \
842     autofs.h     \
843     cacheofs_dir.h \
844     cacheofs_dlog.h \
845     cacheofs_filegrp.h \
846     cacheofs_fs.h \
847     cacheofs_fscache.h \
848     cacheofs_ioctl.h \
849     cacheofs_log.h \
850     decomp.h     \
851     dv_node.h    \
852     sdev_impl.h  \
853     fifonode.h   \

```

```

854     hsfs_isospec.h \
855     hsfs_node.h    \
856     hsfs_rrip.h   \
857     hsfs_spec.h   \
858     hsfs_susp.h   \
859     lofs_info.h   \
860     lofs_node.h   \
861     mntdata.h     \
862     namenode.h    \
863     pc_dir.h      \
864     pc_fs.h       \
865     pc_label.h    \
866     pc_node.h     \
867     pxfs_ki.h     \
868     snode.h       \
869     swapnode.h   \
870     tmp.h         \
871     tmpnode.h    \
872     udf_inode.h   \
873     udf_volume.h  \
874     ufs_acl.h     \
875     ufs_bio.h     \
876     ufs_filio.h  \
877     ufs_fs.h      \
878     ufs_fsdire.h \
879     ufs_inode.h   \
880     ufs_lockfs.h \
881     ufs_log.h     \
882     ufs_mount.h  \
883     ufs_panic.h  \
884     ufs_prot.h   \
885     ufs_quota.h  \
886     ufs_snap.h   \
887     ufs_trans.h  \
888     zfs.h        \
889     zut.h        \

891 PCMCIAHDRS= \
892     pcata.h     \
893     pcser_conf.h \
894     pcser_io.h  \
895     pcser_reg.h \
896     pcser_manuspec.h \
897     pcser_var.h \

899 SCSIHDRS= \
900     scsi.h      \
901     scsi_address.h \
902     scsi_ctl.h  \
903     scsi_fm.h   \
904     scsi_params.h \
905     scsi_pkt.h  \
906     scsi_resource.h \
907     scsi_types.h \
908     scsi_watch.h \

910 SCSSICONFHDRS= \
911     autoconf.h \
912     device.h   \

914 SCSSIGENHDRS= \
915     commands.h \
916     dad_mode.h \
917     inquiry.h  \
918     message.h  \
919     mode.h     \

```

```

920     persist.h      \
921     sense.h        \
922     sff_frames.h   \
923     smp_frames.h   \
924     status.h       \

926 SCIIIMPLHDRS=    \
927     commands.h    \
928     inquiry.h     \
929     mode.h        \
930     scsi_reset_notify.h \
931     scsi_sas.h    \
932     sense.h       \
933     services.h    \
934     smp_transport.h \
935     spc3_types.h  \
936     status.h      \
937     transport.h   \
938     types.h       \
939     uscsi.h       \
940     usmp.h        \

942 SCSTITARGETSHDRS= \
943     ses.h          \
944     sesio.h       \
945     sgendef.h     \
946     stdef.h       \
947     sddef.h       \
948     smp.h         \

950 SCSIADHDRS=

952 SCSCADHDRS=

954 SCIIISCSIHDRS=   \
955     iscsi_door.h  \
956     iscsi_if.h    \

958 SCIVHCIHDRS=     \
959     scsi_vhci.h   \
960     mpapi_impl.h  \
961     mpapi_scsi_vhci.h \

963 SDCARDHDRS=      \
964     sda.h         \
965     sda_impl.h    \
966     sda_ioctl.h   \

968 FC4HDRS=         \
969     fc_transport.h \
970     linkapp.h     \
971     fc.h          \
972     fcp.h         \
973     fcal_transport.h \
974     fcal.h        \
975     fcal_linkapp.h \
976     fcio.h        \

978 FCHDRS=         \
979     fc.h          \
980     fcio.h       \
981     fc_types.h   \
982     fc_appif.h   \

984 FCIMPLHDRS=     \
985     fc_error.h   \

```

```

986     fcph.h

988 FCULPHDRS=       \
989     fcp_util.h    \
990     fcsmd.h       \

992 SATAGENHDRS=     \
993     sata_hba.h    \
994     sata_defs.h   \
995     sata_cfgadm.h \

997 SYSEVENTHDRS=    \
998     ap_driver.h   \
999     dev.h         \
1000    domain.h       \
1001    dr.h           \
1002    env.h          \
1003    eventdefs.h    \
1004    ipmp.h         \
1005    pwrctl.h       \
1006    svm.h          \
1007    vrrp.h         \

1009 CONTRACTHDRS=    \
1010    process.h      \
1011    process_impl.h \
1012    device.h       \
1013    device_impl.h \

1015 USBHDRS=         \
1016    usba.h         \
1017    usbai.h       \

1019 UWBHDRS=         \
1020    uwb.h         \
1021    uwbai.h       \

1023 UWBAHDRS=       \
1024    uwba.h        \

1026 USBAUDHDRS=     \
1027    usb_audio.h   \

1029 USBHUBDHDRS=    \
1030    hub.h         \
1031    hubd_impl.h   \

1033 USBHIDHDRS=     \
1034    hid.h         \

1036 USBHWARCHDRS=   \
1037    hwarc.h       \

1039 USBMSHDRS=      \
1040    usb_bulkonly.h \
1041    usb_cbi.h     \

1043 USBPRNHDRS=     \
1044    usb_printer.h \

1046 USBDCDCHDRS=    \
1047    usb_cdc.h     \

1049 USBVIDHDRS=     \
1050    usbvc.h       \

```

```

1052 USBWCMHDRS= \
1053     usbwcm.h \

1055 UGENHDRS= \
1056     usb_ugen.h \

1058 HOTPLUGHDRS= \
1059     hpcsvc.h \
1060     hpctrl.h \

1062 HOTPLUGPCIHDRS= \
1063     pcicfg.h \
1064     pcihp.h \

1066 RSMHDRS= \
1067     rsm.h \
1068     rsm_common.h \
1069     rsmapi_common.h \
1070     rsmapi.h \
1071     rsmapi_driver.h \
1072     rsmka_path_int.h \

1074 TSOLHDRS= \
1075     label.h \
1076     label_macro.h \
1077     priv.h \
1078     tndb.h \
1079     tsyscall.h \

1081 I1394HDRS= \
1082     cmd1394.h \
1083     id1394.h \
1084     ieee1212.h \
1085     ieee1394.h \
1086     ixl1394.h \
1087     sl394_impl.h \
1088     t1394.h \

1090 # "cmdk" headers used on sparc
1091 SDKTPHDRS= \
1092     dadkio.h \
1093     fdisk.h \

1095 # "cmdk" headers used on i386
1096 DKTPHDRS= \
1097     altsctr.h \
1098     bbh.h \
1099     cm.h \
1100     cmddev.h \
1101     cmdk.h \
1102     cmpkt.h \
1103     controller.h \
1104     dadev.h \
1105     dadk.h \
1106     dadkio.h \
1107     fctypes.h \
1108     fdisk.h \
1109     flowctrl.h \
1110     gda.h \
1111     quetypes.h \
1112     queue.h \
1113     tgcom.h \
1114     tgdk.h \

1116 # "pc" header files used on i386
1117 PCHDRS= \

```

```

1118     avintr.h \
1119     dma_engine.h \
1120     i8272A.h \
1121     pcic_reg.h \
1122     pcic_var.h \
1123     pic.h \
1124     pit.h \
1125     rtc.h \

1127 NXGEHDRS= \
1128     nxge.h \
1129     nxge_common.h \
1130     nxge_common_impl.h \
1131     nxge_defs.h \
1132     nxge_hw.h \
1133     nxge_impl.h \
1134     nxge_ipp.h \
1135     nxge_ipp_hw.h \
1136     nxge_mac.h \
1137     nxge_mac_hw.h \
1138     nxge_fflp.h \
1139     nxge_fflp_hw.h \
1140     nxge_mii.h \
1141     nxge_rxdma.h \
1142     nxge_rxdma_hw.h \
1143     nxge_txc.h \
1144     nxge_txc_hw.h \
1145     nxge_txdma.h \
1146     nxge_txdma_hw.h \
1147     nxge_virtual.h \
1148     nxge_espc.h \

1150 include Makefile.syshdrs

1152 dcam/%.check: dcam/%.h
1153     $(DOT_H_CHECK)

1155 CHECKHDRS= \
1156     $( $(MACH)_HDRS:%.h=%_check) \
1157     $(AUDIOHDRS:%.h=audio/%_check) \
1158     $(AVHDRS:%.h=av/%_check) \
1159     $(BSCHDRS:%.h=%_check) \
1160     $(CHKHDRS:%.h=%_check) \
1161     $(CPUDRVHDRS:%.h=%_check) \
1162     $(CRYPTOHDRS:%.h=crypto/%_check) \
1163     $(DCAMHDRS:%.h=dcam/%_check) \
1164     $(FC4HDRS:%.h=fc4/%_check) \
1165     $(FCHDRS:%.h=fibre-channel/%_check) \
1166     $(FCIMPLHDRS:%.h=fibre-channel/impl/%_check) \
1167     $(FCULPHDRS:%.h=fibre-channel/ulp/%_check) \
1168     $(IBHDRS:%.h=ib/%_check) \
1169     $(IBDHDRS:%.h=ib/clients/ibd/%_check) \
1170     $(IBTLHDRS:%.h=ib/ibtl/%_check) \
1171     $(IBTLIMPLHDRS:%.h=ib/ibtl/impl/%_check) \
1172     $(IBNEXHDRS:%.h=ib/ibnex/%_check) \
1173     $(IBMGTHDRS:%.h=ib/mgt/%_check) \
1174     $(IBMFHDRS:%.h=ib/mgt/ibmf/%_check) \
1175     $(OFHDRS:%.h=ib/clients/of/%_check) \
1176     $(RDMAHDRS:%.h=ib/clients/of/rdma/%_check) \
1177     $(SOL_UVERBSHDRS:%.h=ib/clients/of/sol_uverbs/%_check) \
1178     $(SOL_UCMAHDRS:%.h=ib/clients/of/sol_ucma/%_check) \
1179     $(SOL_OFSHDRS:%.h=ib/clients/of/sol_ofs/%_check) \
1180     $(TAVORHDRS:%.h=ib/adapters/tavor/%_check) \
1181     $(HERMONHDRS:%.h=ib/adapters/hermon/%_check) \
1182     $(MLNXHDRS:%.h=ib/adapters/%_check) \
1183     $(IDMHDRS:%.h=idm/%_check) \

```

```

1184 $(ISCSIHDRS:%.h=iscsi/%.check) \
1185 $(ISCSITHDRS:%.h=iscsit/%.check) \
1186 $(ISOHDRS:%.h=iso/%.check) \
1187 $(FMHDRS:%.h=fm/%.check) \
1188 $(FMFSDHDRS:%.h=fm/fs/%.check) \
1189 $(FMIOHDRS:%.h=fm/io/%.check) \
1190 $(FSDHDRS:%.h=fs/%.check) \
1191 $(LVMHDRS:%.h=lv/%.check) \
1192 $(PCMCIAHDRS:%.h=pcmcia/%.check) \
1193 $(SCSIHDRS:%.h=scsi/%.check) \
1194 $(SCSIADHDRS:%.h=scsi/adapters/%.check) \
1195 $(SCSICONFHDRS:%.h=scsi/conf/%.check) \
1196 $(SCSIIMPLHDRS:%.h=scsi/impl/%.check) \
1197 $(SCSIISCSIHDRS:%.h=scsi/adapters/%.check) \
1198 $(SCSIGHDRS:%.h=scsi/generic/%.check) \
1199 $(SCSITARGETSHDRS:%.h=scsi/targets/%.check) \
1200 $(SCSIVHCIHDRS:%.h=scsi/adapters/%.check) \
1201 $(SATAGENHDRS:%.h=sata/%.check) \
1202 $(SDCARDHDRS:%.h=sdcard/%.check) \
1203 $(SYSEVENTHDRS:%.h=sysevent/%.check) \
1204 $(CONTRACTHDRS:%.h=contract/%.check) \
1205 $(USBBAUDHDRS:%.h=usb/clients/audio/%.check) \
1206 $(USBHUBDHDRS:%.h=usb/hubd/%.check) \
1207 $(USBHIDHDRS:%.h=usb/clients/hid/%.check) \
1208 $(USBHWARDHDRS:%.h=usb/clients/hwarc/%.check) \
1209 $(USBMSHDRS:%.h=usb/clients/mass_storage/%.check) \
1210 $(USBPRNHDRS:%.h=usb/clients/printer/%.check) \
1211 $(USBDCDCHDRS:%.h=usb/clients/usbcdc/%.check) \
1212 $(USBVIDHDRS:%.h=usb/clients/video/usbvc/%.check) \
1213 $(USBWCMHDRS:%.h=usb/clients/usbinput/usbwcm/%.check) \
1214 $(UGENHDRS:%.h=usb/clients/ugen/%.check) \
1215 $(USBHDRS:%.h=usb/%.check) \
1216 $(UWBHDRS:%.h=uwb/%.check) \
1217 $(UWBAHDRS:%.h=uwb/uwba/%.check) \
1218 $(I1394HDRS:%.h=1394/%.check) \
1219 $(RSMHDRS:%.h=rsm/%.check) \
1220 $(TSOLHDRS:%.h=tsol/%.check) \
1221 $(NXGEHDRS:%.h=nxge/%.check)

```

```
1224 .KEEP_STATE:
```

```

1226 .PARALLEL: \
1227 $(CHECKHDRS) \
1228 $(ROOTHDRS) \
1229 $(ROOTAUDHDRS) \
1230 $(ROOTAVHDRS) \
1231 $(ROOTCRYPTOHDRS) \
1232 $(ROOTDCAMHDRS) \
1233 $(ROOTISOHDRS) \
1234 $(ROOTIDMHDRS) \
1235 $(ROOTISCSIHDRS) \
1236 $(ROOTISCSITHDRS) \
1237 $(ROOTFC4HDRS) \
1238 $(ROOTFCHDRS) \
1239 $(ROOTFCIMPLHDRS) \
1240 $(ROOTFCULPHDRS) \
1241 $(ROOTFMHDRS) \
1242 $(ROOTFMIOHDRS) \
1243 $(ROOTFMFSDHDRS) \
1244 $(ROOTFSDHDRS) \
1245 $(ROOTIBDHDRS) \
1246 $(ROOTIBHDRS) \
1247 $(ROOTIBTLHDRS) \
1248 $(ROOTIBTLIMPLHDRS) \
1249 $(ROOTIBNEXHDRS) \

```

```

1250 $(ROOTIBMGTHDRS) \
1251 $(ROOTIBMFHDRS) \
1252 $(ROOTOFHDRS) \
1253 $(ROOTRDMAHDRS) \
1254 $(ROOTSOL_OFSDHDRS) \
1255 $(ROOTSOL_UMADHDRS) \
1256 $(ROOTSOL_UVERBSHDRS) \
1257 $(ROOTSOL_UCMAHDRS) \
1258 $(ROOTTAVORHDRS) \
1259 $(ROOTTHERMONHDRS) \
1260 $(ROOTMLNXHDRS) \
1261 $(ROOTLVMHDRS) \
1262 $(ROOTPCMCIAHDRS) \
1263 $(ROOTSCSIHDRS) \
1264 $(ROOTSCSIADHDRS) \
1265 $(ROOTSCSICONFHDRS) \
1266 $(ROOTSCSIIISCSIHDRS) \
1267 $(ROOTSCSISIGENHDRS) \
1268 $(ROOTSCSIIMPLHDRS) \
1269 $(ROOTSCSIVHCIHDRS) \
1270 $(ROOTSDCARDHDRS) \
1271 $(ROOTSYSEVENTHDRS) \
1272 $(ROOTCONTRACTHDRS) \
1273 $(ROOTUSBHDRS) \
1274 $(ROOTUWBHDRS) \
1275 $(ROOTUWBAHDRS) \
1276 $(ROOTUSBBAUDHDRS) \
1277 $(ROOTUSBHUBDHDRS) \
1278 $(ROOTUSBHIDHDRS) \
1279 $(ROOTUSBHRCCHDRS) \
1280 $(ROOTUSBMSHDRS) \
1281 $(ROOTUSBPRNHDRS) \
1282 $(ROOTUSBDCDCHDRS) \
1283 $(ROOTUSBVIDHDRS) \
1284 $(ROOTUSBWCMHDRS) \
1285 $(ROOTUGENHDRS) \
1286 $(ROOTI1394HDRS) \
1287 $(ROOTHOTPLUGHDRS) \
1288 $(ROOTHOTPLUGPCIHDRS) \
1289 $(ROOTRSMHDRS) \
1290 $(ROOTTSOLHDRS) \
1291 $( $(MACH)_ROOTHDRS)

```

```

1294 install_h: \
1295 $(ROOTDIRS) \
1296 LVMDERIVED_H \
1297 .WAIT \
1298 $(ROOTHDRS) \
1299 $(ROOTAUDHDRS) \
1300 $(ROOTAVHDRS) \
1301 $(ROOTCRYPTOHDRS) \
1302 $(ROOTDCAMHDRS) \
1303 $(ROOTISOHDRS) \
1304 $(ROOTIDMHDRS) \
1305 $(ROOTISCSIHDRS) \
1306 $(ROOTISCSITHDRS) \
1307 $(ROOTFC4HDRS) \
1308 $(ROOTFCHDRS) \
1309 $(ROOTFCIMPLHDRS) \
1310 $(ROOTFCULPHDRS) \
1311 $(ROOTFMHDRS) \
1312 $(ROOTFMFSDHDRS) \
1313 $(ROOTFMIOHDRS) \
1314 $(ROOTFSDHDRS) \
1315 $(ROOTIBDHDRS) \

```

```

1316 $(ROOTIBHDRS) \
1317 $(ROOTIBTLHDRS) \
1318 $(ROOTIBTLIMPLHDRS) \
1319 $(ROOTIBNEXHDRS) \
1320 $(ROOTIBMGTHDRS) \
1321 $(ROOTIBMFHDRS) \
1322 $(ROOTOFHDRS) \
1323 $(ROOTRDMAHDRS) \
1324 $(ROOTSOL_OFSHDRS) \
1325 $(ROOTSOL_UMADHDRS) \
1326 $(ROOTSOL_UVERBSHDRS) \
1327 $(ROOTSOL_UCMAHDRS) \
1328 $(ROOTTAVORHDRS) \
1329 $(ROOTTHERMONHDRS) \
1330 $(ROOTMLNXHDRS) \
1331 $(ROOTLVMHDRS) \
1332 $(ROOTPCMCIAHDRS) \
1333 $(ROOTSCSIHDRS) \
1334 $(ROOTSCSIADHDRS) \
1335 $(ROOTSCSIISCSIIHDRS) \
1336 $(ROOTSCSICONFHDRS) \
1337 $(ROOTSCSIGENHDRS) \
1338 $(ROOTSCSIIMPLHDRS) \
1339 $(ROOTSCSIVHCIHDRS) \
1340 $(ROOTSDCARDHDRS) \
1341 $(ROOTSYSEVENTHDRS) \
1342 $(ROOTCONTRACTHDRS) \
1343 $(ROOTUWBHDRS) \
1344 $(ROOTUWBAHDRS) \
1345 $(ROOTUSBHDRS) \
1346 $(ROOTUSBAUDHDRS) \
1347 $(ROOTUSBHUBDHDRS) \
1348 $(ROOTUSBHIDHDRS) \
1349 $(ROOTUSBHRCHDRS) \
1350 $(ROOTUSBMSHDRS) \
1351 $(ROOTUSBPRNHDRS) \
1352 $(ROOTUSBCDCHDRS) \
1353 $(ROOTUSBVIDHDRS) \
1354 $(ROOTUSBWCMHDRS) \
1355 $(ROOTUGENHDRS) \
1356 $(ROOT1394HDRS) \
1357 $(ROOTHOTPLUGHDRS) \
1358 $(ROOTHOTPLUGPCIIHDRS) \
1359 $(ROOTRSMHDRS) \
1360 $(ROOTTSOLHDRS) \
1361 $( $(MACH)_ROOTHDRS)

```

```
1363 all_h: $(GENHDRS)
```

```
1365 priv_const.h: $(PRIVS_AWK) $(PRIVS_DEF)
```

```
1366 $(AWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v privhfile=$@
```

```
1366 $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v privhfile=$@
```

```
1368 priv_names.h: $(PRIVS_AWK) $(PRIVS_DEF)
```

```
1369 $(AWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v pubhfile=$@
```

```
1369 $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v pubhfile=$@
```

```
1371 usb/usbdevs.h: $(USBDEVS_AWK) $(USBDEVS_DATA)
```

```
1372 $(AWK) -f $(USBDEVS_AWK) $(USBDEVS_DATA) -H > $@
```

```
1372 $(NAWK) -f $(USBDEVS_AWK) $(USBDEVS_DATA) -H > $@
```

```
1374 LVMDERIVED_H:
```

```
1375 cd $(SRC)/uts/common/sys/lvm; pwd; $(MAKE)
```

```
1377 clean:
```

```
1378 $(RM) $(GENHDRS)
```

```
1380 clobber: clean
```

```
1382 check: $(CHECKHDRS)
```

```
1384 FRC:
```

```
1386 # EXPORT DELETE START
```

```
1387 EXPORT_SRC:
```

```
1388 $(RM) wanboot_impl.h+ Makefile+
```

```
1389 sed -e "/EXPORT DELETE START/,/EXPORT DELETE END/d" \
```

```
1390 < wanboot_impl.h > wanboot_impl.h+
```

```
1391 $(MV) wanboot_impl.h+ wanboot_impl.h
```

```
1392 sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
```

```
1393 < Makefile > Makefile+
```

```
1394 $(RM) Makefile
```

```
1395 $(MV) Makefile+ Makefile
```

```
1396 $(CHMOD) 444 Makefile wanboot_impl.h
```

```
1397 # EXPORT DELETE END
```

new/usr/src/uts/i86pc/Makefile.rules

1

14037 Sun May 26 18:24:04 2013
new/usr/src/uts/i86pc/Makefile.rules
XXX Remove awk(1)

```
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) 1992, 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 #
26 # This Makefile defines the build rules for the directory uts/i86pc
27 # and its children. These are the source files which are i86pc
28 # "implementation architecture" dependent.
29 #
30 # The following two-level ordering must be maintained in this file.
31 # Lines are sorted first in order of decreasing specificity based on
32 # the first directory component. That is, i86pc rules come before
33 # intel rules come before common rules.
34 #
35 # Lines whose initial directory components are equal are sorted
36 # alphabetically by the remaining components.
37 #
38 #
39 # Section 1a: C object build rules
40 #
41 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/conf/%.c
42 $(COMPILE.c) -o $@ $<
43 $(CTFCONVERT_O)
44 #
45 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/cpu/amd_opteron/%.c
46 $(COMPILE.c) -o $@ $<
47 $(CTFCONVERT_O)
48 #
49 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/cpu/authenticamd/%.c
50 $(COMPILE.c) -o $@ $<
51 $(CTFCONVERT_O)
52 #
53 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/cpu/generic_cpu/%.c
54 $(COMPILE.c) -o $@ $<
55 $(CTFCONVERT_O)
56 #
57 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/cpu/genuineintel/%.c
58 $(COMPILE.c) -o $@ $<
59 $(CTFCONVERT_O)
60 #
61 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/%.c
```

new/usr/src/uts/i86pc/Makefile.rules

2

```
62 $(COMPILE.c) -o $@ $<
63 $(CTFCONVERT_O)
64 #
65 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/acpi_drv/%.c
66 $(COMPILE.c) -o $@ $<
67 $(CTFCONVERT_O)
68 #
69 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/fipe/%.c
70 $(COMPILE.c) -o $@ $<
71 $(CTFCONVERT_O)
72 #
73 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/acpi/acpidev/%.c
74 $(COMPILE.c) -o $@ $<
75 $(CTFCONVERT_O)
76 #
77 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/acpi/acpinex/%.c
78 $(COMPILE.c) -o $@ $<
79 $(CTFCONVERT_O)
80 #
81 SBD_IOCTL = $(UTSBASE)/i86pc/sys/sbd_ioctl.h
82 DRMACH_IO = $(UTSBASE)/i86pc/io/acpi/drmach_acpi
83 DRMACH_GENERR = $(DRMACH_IO)/sbdgenerr
84 DR_IO = $(UTSBASE)/i86pc/io/dr
85 DR_GENERR = $(DR_IO)/sbdgenerr
86 #
87 $(DRMACH_GENERR): $(DR_IO)/sbdgenerr.pl
88 $(RM) $@
89 $(CAT) $(DR_IO)/sbdgenerr.pl > $@
90 $(CHMOD) +x $@
91 #
92 $(DRMACH_IO)/drmach_err.c: $(DRMACH_GENERR) $(SBD_IOCTL)
93 $(RM) $@
94 $(DRMACH_GENERR) EX86 < $(SBD_IOCTL) > $(DRMACH_IO)/drmach_err.c
95 #
96 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/acpi/drmach_acpi/%.c
97 $(COMPILE.c) -o $@ $<
98 $(CTFCONVERT_O)
99 #
100 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/amd_iommu/%.c
101 $(COMPILE.c) -o $@ $<
102 $(CTFCONVERT_O)
103 #
104 $(DR_GENERR): $(DR_IO)/sbdgenerr.pl
105 $(RM) $@
106 $(CAT) $(DR_IO)/sbdgenerr.pl > $@
107 $(CHMOD) +x $@
108 #
109 $(DR_IO)/dr_err.c: $(DR_GENERR) $(SBD_IOCTL)
110 $(RM) $@
111 $(DR_GENERR) EBSD < $(SBD_IOCTL) > $(DR_IO)/dr_err.c
112 #
113 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/dr/%.c
114 $(COMPILE.c) -o $@ $<
115 $(CTFCONVERT_O)
116 #
117 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/ioat/%.c
118 $(COMPILE.c) -o $@ $<
119 $(CTFCONVERT_O)
120 #
121 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/pci/%.c
122 $(COMPILE.c) -o $@ $<
123 $(CTFCONVERT_O)
124 #
125 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/pciex/%.c
126 $(COMPILE.c) -o $@ $<
127 $(CTFCONVERT_O)
```

```

129 $(OBJSDIR)/%.o: $(UTSBASE)/intel/io/pciex/hotplug/%.c
130 $(COMPILE.c) -o $@ $<
131 $(CTFCONVERT_O)

133 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/pcplusmp/%.c
134 $(COMPILE.c) -o $@ $<
135 $(CTFCONVERT_O)

137 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/pcplusmp/%.s
138 $(COMPILE.s) -o $@ $<

140 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/apix/%.c
141 $(COMPILE.c) -o $@ $<
142 $(CTFCONVERT_O)

144 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/ppm/%.c
145 $(COMPILE.c) -o $@ $<
146 $(CTFCONVERT_O)

148 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/ppm/%.s
149 $(COMPILE.s) -o $@ $<

151 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/psm/%.c
152 $(COMPILE.c) -o $@ $<
153 $(CTFCONVERT_O)

155 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/psm/%.s
156 $(COMPILE.s) -o $@ $<

158 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/tzmon/%.c
159 $(COMPILE.c) -o $@ $<
160 $(CTFCONVERT_O)

162 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/ml/%.s
163 $(COMPILE.s) -o $@ $<

165 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/os/%.c
166 $(COMPILE.c) -o $@ $<
167 $(CTFCONVERT_O)

169 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/os/cpupm/%.c
170 $(COMPILE.c) -o $@ $<
171 $(CTFCONVERT_O)

173 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/boot/%.c
174 $(COMPILE.c) -o $@ $<
175 $(CTFCONVERT_O)

177 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/vm/%.c
178 $(COMPILE.c) -o $@ $<
179 $(CTFCONVERT_O)

181 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/%.c
182 $(COMPILE.c) -o $@ $<
183 $(CTFCONVERT_O)

185 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ppm/%.c
186 $(COMPILE.c) -o $@ $<
187 $(CTFCONVERT_O)

189 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/pciex/%.c
190 $(COMPILE.c) -o $@ $<
191 $(CTFCONVERT_O)

193 $(OBJSDIR)/%.o: $(UTSBASE)/common/os/%.c

```

```

194 $(COMPILE.c) -o $@ $<
195 $(CTFCONVERT_O)

197 $(OBJSDIR)/%.o: $(SRC)/common/dis/i386/%.c
198 $(COMPILE.c) -o $@ $<
199 $(CTFCONVERT_O)

201 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/gfx_private/%.c
202 $(COMPILE.c) -o $@ $<
203 $(CTFCONVERT_O)

205 $(OBJSDIR)/%.o: $(UTSBASE)/i86pc/io/xsvc/%.c
206 $(COMPILE.c) -o $@ $<
207 $(CTFCONVERT_O)

209 $(OBJSDIR)/%.o: $(UTSBASE)/common/xen/os/%.c
210 $(COMPILE.c) -o $@ $<
211 $(CTFCONVERT_O)

213 #
214 # dboot stuff is always 32 bit, linked to run with phys_addr == virt_addr
215 #
216 DBOOT_OBJSDIR = dboot/$(OBJSDIR)
217 DBOOT_MACH_32 = -D_BOOT_TARGET_i386
218 DBOOT_MACH_64 = -D_BOOT_TARGET_amd64
219 DBOOT_DEFS = -D_BOOT $(DBOOT_MACH_$(CLASS))
220 DBOOT_DEFS += -D_MACHDEP -D_KMEMUSER -U_KERNEL -D_I32LPx
221 DBOOT_FLAGS = $(CCVERBOSE) $(CERRWARN) $(CCNOAUTOINLINE)

223 DBOOT_CC_INCL = -I$(SRC)/common $(INCLUDE_PATH)
224 DBOOT_AS_INCL = $(AS_INC_PATH)

226 DBOOT_AS = $(ONBLD_TOOLS)/bin/$(MACH)/aw

228 DBOOT_LINTSDIR = $(DBOOT_OBJSDIR)
229 DBOOT_LINTFLAGS_i86pc = $(LINTFLAGS_i386_32) $(LINTTAGS_i386_32)

231 DBOOT_LINTFLAGS = $(DBOOT_LINTFLAGS_$(PLATFORM)) $(LINTTAGS) $(C99LMODE) \
232 $(CPPFLAGS) $(DBOOT_DEFS)
233 DBOOT_LOCAL_LINTFLAGS = -c -dirout=$(DBOOT_LINTSDIR) -I$(SRC)/common \
234 $(DBOOT_LINTFLAGS) $(DBOOT_LINTTAGS)

236 $(DBOOT_OBJSDIR)/%.o: $(UTSBASE)/i86pc/boot/%.c
237 $(i386_CC) $(DBOOT_FLAGS) -O $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

239 $(DBOOT_OBJSDIR)/%.o: $(UTSBASE)/i86pc/dboot/%.c
240 $(i386_CC) $(DBOOT_FLAGS) -O $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

242 $(DBOOT_OBJSDIR)/%.o: $(UTSBASE)/intel/ia32/%.s
243 $(DBOOT_AS) -P -D_ASM $(DBOOT_DEFS) $(DBOOT_AS_INCL) -o $@ $<

245 $(DBOOT_OBJSDIR)/%.o: $(COMMONBASE)/crypto/shal/%.c
246 $(i386_CC) $(CERRWARN) -O $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

248 $(DBOOT_OBJSDIR)/%.o: $(COMMONBASE)/util/%.c
249 $(i386_CC) $(DBOOT_FLAGS) -O $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

251 $(DBOOT_OBJSDIR)/%.o: $(COMMONBASE)/util/i386/%.s
252 $(DBOOT_AS) -P -D_ASM $(DBOOT_DEFS) $(DBOOT_AS_INCL) -o $@ $<

254 $(DBOOT_OBJSDIR)/%.o: $(UTSBASE)/i86pc/dboot/%.s
255 $(DBOOT_AS) -P -D_ASM $(DBOOT_DEFS) $(DBOOT_AS_INCL) -o $@ $<

257 #
258 # Stuff to build bios_call.o for the kernel.
259 #

```



```

260 MAPFILE_BIOS = $(UTSBASE)/i86pc/conf/Mapfile.bios
261 $(OBJSDIR)/bios_call.o:      $(UTSBASE)/i86pc/ml/bios_call_src.s
262 $(COMPILE.s) -o $(OBJSDIR)/bios_call_src.o \
263 $(UTSBASE)/i86pc/ml/bios_call_src.s
264 $(LD) -dn -M $(MAPFILE_BIOS) \
265 -o $(OBJSDIR)/bios_call_src $(OBJSDIR)/bios_call_src.o
266 @echo ".data" > $(OBJSDIR)/bios_call.s
267 @echo ".globl bios_image" >> $(OBJSDIR)/bios_call.s
268 @echo "bios_image:" >> $(OBJSDIR)/bios_call.s
269 $(ELFEXTRACT) $(OBJSDIR)/bios_call_src >> $(OBJSDIR)/bios_call.s
270 @echo ".align 4" >> $(OBJSDIR)/bios_call.s
271 @echo ".globl bios_size" >> $(OBJSDIR)/bios_call.s
272 @echo "bios_size:" >> $(OBJSDIR)/bios_call.s
273 @echo ".long . - bios_image" >> $(OBJSDIR)/bios_call.s
274 $(COMPILE.s) -o $@ $(OBJSDIR)/bios_call.s

276 #
277 # Stuff to build fb_swtdch.o for the kernel.
278 #
279 MAPFILE_FBSWTDCH = $(UTSBASE)/i86pc/conf/Mapfile.fb_swtdch
280 $(OBJSDIR)/fb_swtdch.o:      $(UTSBASE)/i86pc/ml/fb_swtdch_src.s
281 $(COMPILE.s) -o $(OBJSDIR)/fb_swtdch_src.o \
282 $(UTSBASE)/i86pc/ml/fb_swtdch_src.s
283 $(LD) -dn -M $(MAPFILE_FBSWTDCH) \
284 -o $(OBJSDIR)/fb_swtdch_src $(OBJSDIR)/fb_swtdch_src.o
285 @echo ".data" > $(OBJSDIR)/fb_swtdch.s
286 @echo ".globl fb_swtdch_image" >> $(OBJSDIR)/fb_swtdch.s
287 @echo "fb_swtdch_image:" >> $(OBJSDIR)/fb_swtdch.s
288 $(ELFEXTRACT) $(OBJSDIR)/fb_swtdch_src >> $(OBJSDIR)/fb_swtdch.s
289 @echo ".align 4" >> $(OBJSDIR)/fb_swtdch.s
290 @echo ".globl fb_swtdch_size" >> $(OBJSDIR)/fb_swtdch.s
291 @echo "fb_swtdch_size:" >> $(OBJSDIR)/fb_swtdch.s
292 @echo ".long . - fb_swtdch_image" >> $(OBJSDIR)/fb_swtdch.s
293 $(COMPILE.s) -o $@ $(OBJSDIR)/fb_swtdch.s

295 # ridiculous contortions ---
296 ATOMIC_SUBDIR_32 = i386
297 ATOMIC_SUBDIR_64 = amd64
298 ATOMIC_SUBDIR = $(ATOMIC_SUBDIR_$(CLASS))

300 $(OBJSDIR)/%.o:      $(SRC)/common/atomic/$(ATOMIC_SUBDIR)/%.s
301 $(COMPILE.s) -o $@ <

303 #
304 # dtrace stubs
305 #

307 $(OBJSDIR)/dtracestubs.s:      $(UNIX_O) $(LIBS)
308 $(NM) -u $(UNIX_O) $(LIBS) | $(GREP) __dtrace_probe_ | $(SORT) | \
309 $(UNIQ) | $(AWK) '{ \
310 $(UNIQ) } $(NAWK) '{ \
311 printf("\t.globl %s\n\t.type %s,@function\n%s:\n", \
312 $1, $1, $1); }' > $(OBJSDIR)/dtracestubs.s

313 $(DTRACESTUBS): $(DTRACESTUBS_O)
314 $(BUILD.SO) $(DTRACESTUBS_O)

316 #
317 # Section lb: Lint 'object' build rules
318 #
319 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/conf/%.c
320 @$(LHEAD) $(LINT.c) < $(LTAIL)

322 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/cpu/amd_opteron/%.c
323 @$(LHEAD) $(LINT.c) < $(LTAIL)

```

```

325 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/cpu/authenticamd/%.c
326 @$(LHEAD) $(LINT.c) < $(LTAIL)

328 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/cpu/generic_cpu/%.c
329 @$(LHEAD) $(LINT.c) < $(LTAIL)

331 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/cpu/genuineintel/%.c
332 @$(LHEAD) $(LINT.c) < $(LTAIL)

334 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/%.c
335 @$(LHEAD) $(LINT.c) < $(LTAIL)

337 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/acpi_drv/%.c
338 @$(LHEAD) $(LINT.c) < $(LTAIL)

340 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/fipe/%.c
341 @$(LHEAD) $(LINT.c) < $(LTAIL)

343 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/acpi/acpidev/%.c
344 @$(LHEAD) $(LINT.c) < $(LTAIL)

346 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/acpi/acpinex/%.c
347 @$(LHEAD) $(LINT.c) < $(LTAIL)

349 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/acpi/drmach_acpi/%.c
350 @$(LHEAD) $(LINT.c) < $(LTAIL)

352 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/amd_iommu/%.c
353 @$(LHEAD) $(LINT.c) < $(LTAIL)

355 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/dr/%.c
356 @$(LHEAD) $(LINT.c) < $(LTAIL)

358 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/ioat/%.c
359 @$(LHEAD) $(LINT.c) < $(LTAIL)

361 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/pci/%.c
362 @$(LHEAD) $(LINT.c) < $(LTAIL)

364 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/pciex/%.c
365 @$(LHEAD) $(LINT.c) < $(LTAIL)

367 $(LINTSDIR)/%.ln:      $(UTSBASE)/intel/io/pciex/hotplug/%.c
368 @$(LHEAD) $(LINT.c) < $(LTAIL)

370 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/pcplusmp/%.c
371 @$(LHEAD) $(LINT.c) < $(LTAIL)

373 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/pcplusmp/%.s
374 @$(LHEAD) $(LINT.s) < $(LTAIL)

376 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/apix/%.c
377 @$(LHEAD) $(LINT.c) < $(LTAIL)

379 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/apix/%.s
380 @$(LHEAD) $(LINT.s) < $(LTAIL)

382 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/ppm/%.c
383 @$(LHEAD) $(LINT.c) < $(LTAIL)

385 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/ppm/%.s
386 @$(LHEAD) $(LINT.s) < $(LTAIL)

388 $(LINTSDIR)/%.ln:      $(UTSBASE)/i86pc/io/psm/%.c
389 @$(LHEAD) $(LINT.c) < $(LTAIL)

```

new/usr/src/uts/i86pc/Makefile.rules

7

```

391 $(LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/io/psm/%.s
392     @$(LHEAD) $(LINT.s) $< $(LTAIL))

394 $(LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/io/tzmon/%.c
395     @$(LHEAD) $(LINT.c) $< $(LTAIL))

397 $(LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/ml/%.s
398     @$(LHEAD) $(LINT.s) $< $(LTAIL))

400 $(LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/os/%.c
401     @$(LHEAD) $(LINT.c) $< $(LTAIL))

403 $(LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/os/cpum/%.c
404     @$(LHEAD) $(LINT.c) $< $(LTAIL))

406 $(LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/boot/%.c
407     @$(LHEAD) $(LINT.c) $< $(LTAIL))

409 $(LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/vm/%.c
410     @$(LHEAD) $(LINT.c) $< $(LTAIL))

412 $(LINTS_DIR)/%.ln: $(UTSBASE)/common/io/%.c
413     @$(LHEAD) $(LINT.c) $< $(LTAIL))

415 $(LINTS_DIR)/%.ln: $(UTSBASE)/common/io/ppm/%.c
416     @$(LHEAD) $(LINT.c) $< $(LTAIL))

418 $(LINTS_DIR)/%.ln: $(UTSBASE)/common/io/pciex/%.c
419     @$(LHEAD) $(LINT.c) $< $(LTAIL))

421 $(LINTS_DIR)/%.ln: $(UTSBASE)/common/os/%.c
422     @$(LHEAD) $(LINT.c) $< $(LTAIL))

424 $(LINTS_DIR)/%.ln: $(SRC)/common/dis/i386/%.c
425     @$(LHEAD) $(LINT.c) $< $(LTAIL))

427 $(LINTS_DIR)/%.ln: $(SRC)/common/atomic/%.c
428     @$(LHEAD) $(LINT.c) $< $(LTAIL))

430 $(LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/io/gfx_private/%.c
431     @$(LHEAD) $(LINT.c) $< $(LTAIL))

433 $(LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/io/xsvc/%.c
434     @$(LHEAD) $(LINT.c) $< $(LTAIL))

436 $(LINTS_DIR)/%.ln: $(UTSBASE)/common/xen/os/%.c
437     @$(LHEAD) $(LINT.c) $< $(LTAIL))

439 #
440 # bios call has a funky name change while building
441 #
442 $(LINTS_DIR)/bios_call.ln: $(UTSBASE)/i86pc/ml/bios_call_src.s
443     @$(LHEAD) $(LINT.s) $(UTSBASE)/i86pc/ml/bios_call_src.s $(LTAIL))
444     @mv $(LINTS_DIR)/bios_call_src.ln $(LINTS_DIR)/bios_call.ln

446 #
447 # So does fb_swtdch
448 #
449 $(LINTS_DIR)/fb_swtdch.ln: $(UTSBASE)/i86pc/ml/fb_swtdch_src.s
450     @$(LHEAD) $(LINT.s) $(UTSBASE)/i86pc/ml/fb_swtdch_src.s $(LTAIL))
451     @mv $(LINTS_DIR)/fb_swtdch_src.ln $(LINTS_DIR)/fb_swtdch.ln

453 #
454 # dboot always compiles in 32-bit mode, so force lint to be 32-bit mode too.
455 #
456 $(DBOOT_LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/dboot/%.c

```

new/usr/src/uts/i86pc/Makefile.rules

8

```

457     @$(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

459 $(DBOOT_LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/dboot/%.s
460     @$(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

462 $(DBOOT_LINTS_DIR)/%.ln: $(COMMONBASE)/crypto/shal/%.c
463     @$(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

465 $(DBOOT_LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/boot/%.c
466     @$(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

468 $(DBOOT_LINTS_DIR)/%.ln: $(COMMONBASE)/util/%.c
469     @$(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

471 $(DBOOT_LINTS_DIR)/%.ln: $(COMMONBASE)/util/i386/%.s
472     @$(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

```

new/usr/src/uts/intel/Makefile

1

```
*****
5558 Sun May 26 18:24:05 2013
new/usr/src/uts/intel/Makefile
XXX Remove nawk(1)
*****
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 # uts/intel/Makefile
22 #
23 # Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 # This makefile drives the production of all implementation architecture
26 # independent modules for Intel processors.

28 UTSBASE = ..

30 include Makefile.intel

32 LINT_KMODS_X1 = $(LINT_KMODS:nsmb=)
33 LINT_KMODS_X2 = $(LINT_KMODS_X1:smbfs=)
34 LINT_KMODLIBS = $(LINT_KMODS_X2:e1000g=)
35 LINT_LIBS = $(LINT_LIB) $(GEN_LINT_LIB) \
36             $(LINT_KMODLIBS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
37             $(CLOSED_LINT_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln)

39 # EXPORT DELETE START
40 $(CLOSED_BUILD)LINT_LIBS += $(SVVS_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln)
41 $(CLOSED_BUILD)LINT_CLOSED_XMOD4 = $(CLOSED_XMODS:bnx=)
42 $(CLOSED_BUILD)LINT_CLOSED_XMOD3 = $(LINT_CLOSED_XMOD4:bnxe=)
43 $(CLOSED_BUILD)LINT_CLOSED_XMOD2 = $(LINT_CLOSED_XMOD3:lsimega=)
44 $(CLOSED_BUILD)LINT_CLOSED_XMOD1 = $(LINT_CLOSED_XMOD2:adpu320=)
45 $(CLOSED_BUILD)LINT_LIBS += $(LINT_XMODLIBS:%=$(LINT_LIB_DIR)/llib-1%.ln)

47 #
48 # dprov is delivered in the SUNWcryptoint package.
49 #
50 DRV_KMODS += dprov

52 # EXPORT DELETE END

54 #
55 #
56 def := TARGET= def
57 def.prereq := TARGET= def
58 all := TARGET= all
59 all.prereq := TARGET= all
60 install := TARGET= install
61 install.prereq := TARGET= all
```

new/usr/src/uts/intel/Makefile

2

```
62 clean := TARGET= clean
63 clobber := TARGET= clobber
64 lint := TARGET= lint
65 lint.prereq := TARGET= lint
66 modlintlib := TARGET= modlintlib
67 modlist := TARGET= modlist
68 modlist := NO_STATE= -K $$MODSTATE$$$$
69 clean.lint := TARGET= clean.lint
70 check := TARGET= check
71 install_h := TARGET= install_h
72 install_h.prereq := TARGET= install_h

74 .KEEP_STATE:

76 .PARALLEL: $(PARALLEL_KMODS) $(CLOSED_KMODS) $(SVVS) $(XMODS) \
77            $(CLOSED_XMODS) config $(LINT_DEPS)

79 def all install clean clobber modlist: $(KMODS) $(CLOSED_KMODS) \
80            $(SVVS) $(XMODS) $(CLOSED_XMODS) config

83 #
84 # Privilege constants
85 #
86 # NOTE: The rules for generating priv_const.c file are shared between all
87 # processor architectures and should be kept in sync. If they are changed in
88 # this file make sure that sparc rules are updated as well.
89 #
90 PRIVS_C = $(SRC)/uts/common/os/priv_const.c

92 $(PRIVS_C): $(PRIVS_AWK) $(PRIVS_DEF)
93             $(AWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) cfile=$@
94             $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) cfile=$@

95 #
96 # Prerequisites
97 #
98 # The uts/Makefile defines build parallelism for x86 platforms such that i86pc,
99 # i86xp and intel are all built in parallel. This requires building certain
100 # parts before the parallel build can start. The uts/Makefile appends the
101 # '.prereq' string to the original target and executes this Makefile to build
102 # any prerequisites needed before the full parallel build can start. After that
103 # make continues with normal targets.
104 #
105 # Any build prerequisites for x86 builds should be described here.
106 #
107 # genassym is used to build intel/dtrace and genunix, so it should be built
108 # first.
109 #
110 # priv_const.c is required to build genunix.
111 #
112 # genunix is used by everyone to ctf-merge with. Genunix is CTF-merged with
113 # intel/ip so as a side effect this dependency builds intel/ip as part of the
114 # prerequisites.
115 #
116 # intel/dtrace depends on i86pc/genassym, so we need to build both
117 # i86pc/genassym and intel/genassym.
118 #
119 all.prereq install.prereq def.prereq: genunix FRC
120             @cd ../i86pc/genassym; pwd; $(MAKE) $(@:%.prereq=%)

122 #
123 # i86pc lint libraries should be built first
124 #
125 lint.prereq: FRC
126             @cd ../i86pc; pwd; $(MAKE) $(NO_STATE) lint
```

```

128 #
129 # Nothing to do for any other prerequisite targets.
130 #
131 %.prereq:
132
133 genunix: $(PRIVS_C)
134
135 modlintlib clean.lint: $(LINT_KMODS) $(CLOSED_LINT_KMODS) $(SVVS) \
136     $(XMODS) $(CLOSED_XMODS)
137
138 $(KMODS) $(SUBDIRS) config:      FRC
139     @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)
140
141 $(CLOSED_KMODS):                FRC
142     cd $(CLOSED)/uts/intel/$@; pwd; $(MAKE) $(NO_STATE) $(TARGET)
143
144 $(XMODS):                       FRC
145     @if [ -f $@/Makefile ]; then \
146         cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
147     else \
148         true; \
149     fi
150
151 $(SVVS) $(CLOSED_XMODS):        FRC
152     @if [ -f $(CLOSED)/uts/intel/$@/Makefile ]; then \
153         cd $(CLOSED)/uts/intel/$@; pwd; \
154         $(MAKE) $(NO_STATE) $(TARGET); \
155     else \
156         true; \
157     fi
158
159 install_h check:              FRC
160     @cd sys; pwd; $(MAKE) $(TARGET)
161     @cd asm; pwd; $(MAKE) $(TARGET)
162     @cd ia32/sys; pwd; $(MAKE) $(TARGET)
163     @cd amd64/sys; pwd; $(MAKE) $(TARGET)
164
165 #
166 # Work-around to disable acpica global crosscheck lint warnings
167 #
168 LGREP.intel = grep -v 'intel/io/acpica'
169
170 #
171 # Full kernel lint target.
172 #
173 LINT_TARGET = globalint
174
175 # workaround for multiply defined errors
176 globalint := LINTFLAGS += -erroff=E_NAME_MULTIPLY_DEF2
177
178 globalint:
179     @pwd
180     @-$(ECHO) "\nFULL KERNEL: global crosschecks:"
181     @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.intel) | $(LGREP.2)
182
183 lint: modlintlib .WAIT $(LINT_DEPS)
184
185 # EXPORT DELETE START
186
187 EXPORT_SRC:
188     $(RM) Makefile+
189     sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \
190         < Makefile > Makefile+
191     $(MV) Makefile+ Makefile
192     $(CHMOD) 444 Makefile

```

```

194 # EXPORT DELETE END
195
196 include ../Makefile.targ

```

new/usr/src/uts/intel/config/Makefile

1

```
*****
3485 Sun May 26 18:24:05 2013
new/usr/src/uts/intel/config/Makefile
XXX Remove nawk(1)
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # uts/intel/config/Makefile
23 # Copyright 2010 Nexenta Systems, Inc. All rights reserved.
24 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 # This makefile drives the production of the kernel configuration files.
28 #
29 # intel architecture dependent
30 #
31 #
32 #
33 # Path to the base of the uts directory tree (usually /usr/src/uts).
34 #
35 UTBASE = ../..
36 #
37 #
38 # Include common rules.
39 #
40 include $(UTSBASE)/intel/Makefile.intel
41 #
42 #
43 # Define the module and object file sets.
44 #
45 SRCDIR = $(UTSBASE)/intel/os
46 PATH2INST = path_to_inst
47 CONFIGFILES = name_to_major_minor_perm driver_aliases name_to_sysnum \
48 driver_classes dacf.conf $(PATH2INST) mach
49 SECCONFIG = device_policy priv_names
50 BOOTIMAGE = splashimage.xpm
51 ALTBOOTIMAGE = solaris.xpm # Temporary, for compatibility
52 BOOTENVRC = bootenv.rc
53 DEVMASTER = master
54 FILEMODE = 644
55 ROOTETC = $(ROOT)/etc
56 ROOTETCSEC = $(ROOTETC)/security
57 ROOTBOOT = $(ROOT)/boot
58 BOOTSOLARIS = $(ROOTBOOT)/solaris
59 BOOTDEVICEDB = $(BOOTSOLARIS)/devicedb
60 #
61 SRCFILES = $(CONFIGFILES:%=$(SRCDIR)/%) $(SECCONFIG:%=$(SRCDIR)/%)
```

new/usr/src/uts/intel/config/Makefile

2

```
62 SRCFILES += $(BOOTIMAGE:%=$(SRCDIR)/%)
63 SRCFILES += $(BOOTENVRC:%=$(SRCDIR)/%)
64 SRCFILES += $(DEVMASTER:%=$(SRCDIR)/%)
65 ROOTCONFIGFILES = $(CONFIGFILES:%=$(ROOTETC)/%) $(SECCONFIG:%=$(ROOTETCSEC)/%)
66 ROOTCONFIGFILES += $(BOOTIMAGE:%=$(ROOTBOOT)/%)
67 ROOTCONFIGFILES += $(BOOTENVRC:%=$(BOOTSOLARIS)/%)
68 ROOTCONFIGFILES += $(DEVMASTER:%=$(BOOTDEVICEDB)/%)
69 ROOTALTBOOTIMAGE = $(ALTBOOTIMAGE:%=$(ROOTBOOT)/%)
70 #
71 #
72 # Default build targets.
73 #
74 .KEEP_STATE:
75 #
76 def all: $(SRCFILES)
77 #
78 install: all $(ROOTCONFIGFILES) $(ROOTALTBOOTIMAGE)
79 #
80 #
81 # The ROOTETC directory is made by ../../../Targetdirs
82 # through the rootdirs target in ../../../Makefile.
83 #
84 # etc/path_to_inst has permissions different from the other config files
85 #
86 $(ROOTETC)/$(PATH2INST):= FILEMODE=444
87 #
88 $(ROOTETC)/%: $(SRCDIR)/%
89 $(INS.file)
90 #
91 $(ROOTETCSEC)/%: $(SRCDIR)/%
92 $(INS.file)
93 #
94 $(ROOTBOOT):
95 $(INS) -s -d -m $(DIRMODE) $@
96 #
97 $(ROOTBOOT)/%: $(ROOTBOOT) $(SRCDIR)/%
98 $(INS.file)
99 #
100 $(ROOTBOOT)/solaris.xpm: $(ROOTBOOT)/splashimage.xpm
101 $(RM) $@; $(LN) $(ROOTBOOT)/splashimage.xpm $@
102 #
103 #
104 $(BOOTSOLARIS):
105 $(INS) -s -d -m $(DIRMODE) $@
106 #
107 $(BOOTSOLARIS)/%: $(BOOTSOLARIS) $(SRCDIR)/%
108 $(INS.file)
109 #
110 $(BOOTDEVICEDB):
111 $(INS) -s -d -m $(DIRMODE) $@
112 #
113 $(BOOTDEVICEDB)/%: $(BOOTDEVICEDB) $(SRCDIR)/%
114 $(INS.file)
115 #
116 $(SRCDIR)/priv_names: $(PRIVS_DEF) $(PRIVS_AWK)
117 $(AWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v pnamesfile=$@
118 $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v pnamesfile=$@
119 clean clobber lint:
120 #
121 #
122 # No lints, so keep the clobber and clean targets reasonable.
123 #
124 CLEANLINTFILES = Nothing_to_remove
125 CLEANFILES = Nothing_to_remove
126 CLOBBERFILES = Nothing_to_remove
```

new/usr/src/uts/intel/config/Makefile

3

```
128 #  
129 #     Include common targets.  
130 #  
131 include $(UTSBASE)/intel/Makefile.targ
```

new/usr/src/uts/sparc/config/Makefile

1

```
*****
2441 Sun May 26 18:24:06 2013
new/usr/src/uts/sparc/config/Makefile
XXX Remove awk(1)
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # uts/sparc/config/Makefile
23 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # This makefile drives the production of the kernel configuration files.
27 #
28 # sparc architecture dependent
29 #
30 #
31 #
32 # Path to the base of the uts directory tree (usually /usr/src/uts).
33 #
34 UTSBASE = ../..
35 #
36 #
37 # Include common rules.
38 #
39 include $(UTSBASE)/sparc/Makefile.sparc
40 #
41 #
42 # Define the module and object file sets.
43 #
44 SRCDIR = $(UTSBASE)/sparc/os
45 PATH2INST = path_to_inst
46 CONFIGFILES = name_to_major minor_perm driver_aliases name_to_sysnum \
47 driver_classes dacf.conf $(PATH2INST)
48 SECCONFIG = device_policy priv_names
49 FILEMODE = 644
50 ROOTETC = $(ROOT)/etc
51 ROOTETCSEC = $(ROOTETC)/security
52 #
53 SRCFILES = $(CONFIGFILES:%=$(SRCDIR)/%) $(SECCONFIG:%=$(SRCDIR)/%)
54 ROOTCONFIGFILES = $(CONFIGFILES:%=$(ROOTETC)/%) $(SECCONFIG:%=$(ROOTETCSEC)/%)
55 #
56 #
57 # Default build targets.
58 #
59 .KEEP_STATE:
60 #
61 def all: $(SRCFILES)
```

new/usr/src/uts/sparc/config/Makefile

2

```
63 install: all $(ROOTCONFIGFILES)
64 #
65 #
66 # The ROOTETC directory is made by ../../../Targetdirs
67 # through the rootdirs target in ../../../Makefile.
68 #
69 # etc/path_to_inst has permissions different from the other config files
70 #
71 $(ROOTETC)/$(PATH2INST):= FILEMODE=444
72 #
73 $(ROOTETC)/%: $(SRCDIR)/%
74 $(INS.file)
75 #
76 $(ROOTETCSEC)/%: $(SRCDIR)/%
77 $(INS.file)
78 #
79 $(SRCDIR)/priv_names: $(PRIVS_DEF) $(PRIVS_AWK)
80 $(AWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v pnamesfile=$@
80 $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v pnamesfile=$@
81 #
82 clean clobber lint:
83 #
84 #
85 # No lints, so keep the clobber and clean targets reasonable.
86 #
87 CLEANLINTFILES = Nothing_to_remove
88 CLEANFILES = Nothing_to_remove
89 CLOBBERFILES = Nothing_to_remove
90 #
91 #
92 # Include common targets.
93 #
94 include $(UTSBASE)/sparc/Makefile.targ
```

new/usr/src/uts/sun4u/Makefile

1

```
*****
8966 Sun May 26 18:24:06 2013
new/usr/src/uts/sun4u/Makefile
XXX Remove awk(1)
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # This makefile drives the production of all implementation architecture
26 # dependent modules for the sun4u architecture.
27 #
28 #
29 UTSBASE = ..
30 #
31 include Makefile.sun4u
32 #
33 # The following are SPARC specific (rather than sun4u) specific modules
34 # which are required for the sun4u kernel to completely lint. They are
35 # not involved in the build in any other way. In order to minimize
36 # build time, it is assumed that they are up to date. But since sun4u
37 # is really a separate architecture we cannot use the v7 sparc modules.
38 #
39 #
40 SPARC_LIB_DIR = $(UTSBASE)/sparc/lint-libs/$(OBSJ_DIR)
41 #
42 SPARC_LINTS =
43 #
44 #
45 #
46 #
47 LINT_LIBS = $(LINT_LIB) \
48             $(LINT_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
49             $(CLOSED_LINT_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
50             $(SPARC_LINTS:%=$(SPARC_LIB_DIR)/llib-1%.ln)
51 #
52 #
53 def := TARGET= def
54 def.prereq := TARGET= def
55 all := TARGET= all
56 all.prereq := TARGET= all
57 install := TARGET= install
58 install.prereq := TARGET= all
59 install_h := TARGET= install_h
60 install_h.prereq := TARGET= install_h
61 clean := TARGET= clean
```

new/usr/src/uts/sun4u/Makefile

2

```
62 clobber := TARGET= clobber
63 lint := TARGET= lint
64 lint.prereq := TARGET= lint
65 lintlib := TARGET= lintlib
66 modlintlib := TARGET= modlintlib
67 modlist := TARGET= modlist
68 modlist modlist.sparc := NO_STATE= -K $$MODSTATE$$$$
69 clean.lint := TARGET= clean.lint
70 check := TARGET= check
71 #
72 .KEEP_STATE:
73 #
74 .PARALLEL: $(PARALLEL_KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
75             $(IMPLEMENTATIONS) $(CLOSED_IMPLEMENTATIONS) \
76             modlist modlist.sparc
77 #
78 # Override for CPU_KMODS... they cannot be built
79 # in parallel
80 .NO_PARALLEL: $(CPU_KMODS)
81 #
82 def all clean clobber clean.lint: genassym unix .WAIT \
83             $(KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
84             $(IMPLEMENTATIONS) $(CLOSED_IMPLEMENTATIONS)
85 #
86 # list the modules under sun4u.
87 modlist: unix $(KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
88             $(IMPLEMENTATIONS: .WAIT=) $(CLOSED_IMPLEMENTATIONS)
89 #
90 # list the modules for Install -k sun4u.
91 modlist.karch: modlist modlist.sparc
92 #
93 modlist.sparc:
94     @cd $(SRC)/uts/sparc; pwd; $(MAKE) $(NO_STATE) modlist
95 #
96 install: install_platforms genassym unix .WAIT $(KMODS) $(CLOSED_KMODS) \
97             $(XMODS) $(CLOSED_XMODS) $(IMPLEMENTATIONS) $(CLOSED_IMPLEMENTATIONS)
98 #
99 lintlib: unix
100 #
101 modlintlib: $(LINT_KMODS) $(CLOSED_LINT_KMODS)
102 #
103 genassym unix $(KMODS): FRC
104     @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)
105 #
106 #
107 # Privilege constants
108 #
109 # NOTE: The rules for generating priv_const.c file are shared between all
110 # processor architectures and should be kept in sync. If they are changed in
111 # this file make sure that x86 rules are updated as well.
112 #
113 PRIVS_C = $(UTSBASE)/common/os/priv_const.c
114 #
115 $(PRIVS_C): $(PRIVS_AWK) $(PRIVS_DEF)
116     $(AWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) cfile=$@
117     $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) cfile=$@
118 #
119 #
120 # Prerequisites
121 #
122 # The uts/Makefile defines build parallelism for sun4 platforms such that sparc,
123 # sun4u and sun4v are all built in parallel. Also this Makefile specifies that
124 # all IMPLEMENTATIONS sun4u sub-platforms are built in parallel. This requires
125 # building certain parts before the parallel build can start. The uts/Makefile
126 # appends the '.prereq' string to the original target and executes this Makefile
```



```

127 # to build any prerequisites needed before the full parallel build can start.
128 # After that make continues with normal targets.
129 #
130 # Any build prerequisites for sun4 and IMPLEMENTATIONS builds should be
131 # described here.
132 #
133 # genassym is used to build dtrace and genunix, so it should be built first.
134 #
135 # priv_const.c is required to build genunix.
136 #
137 # genunix is used by everyone to ctfmerge with. Genunix is merged with sparc/ip
138 # so as a side effect this dependency builds sparc/ip as part of the
139 # prerequisites.
140 #
141 # unix is not required by itself but several sun4u platforms require
142 # sun4u/platmod to be present. The easiest way to achieve this is to build
143 # sun4u/unix first since sun4u/unix Makefile builds sun4u/platform correctly.
144 # This causes full sun4u/unix to be built before all sun4u platforms and
145 # before uts/sun4v and uts/sparc, but it acceptable since it is not spending
146 # too much time building sun4u/unix.
147 #
148 all.prereq def.prereq install.prereq: genassym genunix unix

150 #
151 # Various sun4u platforms expect proto/root_sparc/platform/sun4u/include to be
152 # present. This is handled by running make install_h in sun4u/unix directory
153 # first.
154 #
155 install_h.prereq: FRC
156 @cd sys; pwd; $(MAKE) $(TARGET)

158 #
159 # sun4u/unix and sun4u/genunix should be linted first since sparc does global
160 # cross-check with these lint libraries. The sun4u/unix and sun4u/genunix can be
161 # linted in parallel.
162 #
163 LINT_PREREQ = unix.lint genunix.lint
164 lint.prereq: $(LINT_PREREQ)

166 .PARALLEL: $(LINT_PREREQ)

168 $(LINT_PREREQ):
169 @cd %(:%.lint=%); pwd; $(MAKE) $(TARGET)

171 #
172 # Nothing to do with any other prerequisites
173 #
174 %.prereq:

176 #
177 # Platform inter-dependencies
178 #
179 lw8: serengeti

181 quasar: darwin

183 #
184 # The genunix requires priv_const.c file to be generated first.
185 #
186 genunix: $(PRIVS_C)

188 #
189 # Rules
190 #

192 $(IMPLEMENTATIONS): FRC

```

```

193 @cd %@; pwd; THISIMPL=$@ $(MAKE) $(NO_STATE) $(TARGET)

195 $(CLOSED_IMPLEMENTATIONS): FRC
196 cd $(CLOSED)/uts/sun4u/%@; pwd; \
197 THISIMPL=$@ $(MAKE) $(NO_STATE) $(TARGET); \

199 $(XMODS): FRC
200 @if [ -f %@/Makefile ]; then \
201 cd %@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
202 else \
203 true; \
204 fi

206 $(CLOSED_XMODS): FRC
207 @if [ -f $(CLOSED)/uts/sun4u/%@/Makefile ]; then \
208 cd $(CLOSED)/uts/sun4u/%@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
209 else \
210 true; \
211 fi

213 $(CLOSED_KMODS): FRC
214 cd $(CLOSED)/uts/sun4u/%@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

216 install_h check: install_platforms $(IMPLEMENTATIONS) \
217 $(CLOSED_IMPLEMENTATIONS) FRC
218 @cd sys; pwd; $(MAKE) $(TARGET)
219 @cd vm; pwd; $(MAKE) $(TARGET)

221 #
222 # Rules for the /platforms directories. This is hardwired here because
223 # the first stage of the project (KBI) only implements the userland
224 # changes, but the only reasonable place to record the aliases is
225 # here in kernel land.
226 #
227 $(ROOT_PLAT_DIRS): $(ROOT_PLAT_DIR)
228 -$(INS.dir)

230 #
231 # create directories in /usr/platform/ for the implementations that are
232 # defined in $(IMPLEMENTED_PLATFORM)
233 # (eg. SUNW,Ultra-1)
234 #
235 # Foreach $(IMPLEMENTED_PLATFORM) there can be a list of $(LINKED_PLATFORMS)
236 # that are linked to it.
237 #
238 $(USR_PLAT_DIR)/$(IMPLEMENTED_PLATFORM): $(USR_PLAT_DIR)
239 -$(INS.dir)

241 #
242 # create the links in /usr/platform/ foreach $(LINKED_PLATFORMS)
243 # to it's corresponding $(IMPLEMENTED_PLATFORM).
244 #
245 PLATFORMS = $(LINKED_PLATFORMS)

247 $(USR_PLAT_DIRS): $(USR_PLAT_DIR)
248 $(INS.slink3)

250 PLATFORMS += $(IMPLEMENTED_PLATFORM)

252 #
253 # Make the /platforms directories. This is hardwired here because
254 # the first stage of the project (KBI) only implements the userland
255 # changes, but the only reasonable place to record the aliases is
256 # here in kernel land.
257 #
258 install_platforms: $(ROOT_PSM_DIR) $(USR_PSM_DIR) \

```

new/usr/src/uts/sun4u/Makefile

5

```
259 $(ROOT_PLAT_DIRS) $(USR_PLAT_DIRS) \  
260 $(USR_DESKTOP_DIR) $(USR_DESKTOP_INC_DIR) \  
261 $(USR_DESKTOP_SBIN_DIR) $(USR_DESKTOP_LIB_DIR)  
  
263 #  
264 # rules for making include, sbin, lib dirs/links in  
265 # /usr/platform/$(PLATFORM)/ for desktop platforms  
266 #  
267 $(USR_DESKTOP_INC_DIR):      $(USR_DESKTOP_DIR)  
268     $(INS.slink4)  
  
270 $(USR_DESKTOP_SBIN_DIR):     $(USR_DESKTOP_DIR)  
271     $(INS.slink5)  
  
273 $(USR_DESKTOP_LIB_DIR):     $(USR_DESKTOP_DIR)  
274     -$(INS.dir)  
  
276 #  
277 #     Full kernel lint target.  
278 #  
279 LINT_TARGET      = globallint  
  
281 globallint:  
282     @pwd  
283     @-$(ECHO) "\nSUN4U KERNEL: global crosschecks:"  
284     @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.2)  
  
286 lint:    lintlib .WAIT modlintlib .WAIT $(SPARC_LINTS) $(LINT_DEPS) \  
287           $(IMPLEMENTATIONS) $(CLOSED_IMPLEMENTATIONS) $(CPU_KMODS)  
  
289 # EXPORT DELETE START  
  
291 EXPORT_SRC:  
292     $(RM) Makefile+  
293     sed -e "/^# EXPORT DELETE START/,/^# EXPORT DELETE END/d" \  
294         < Makefile > Makefile+  
295     $(MV) Makefile+ Makefile  
296     $(CHMOD) 444 Makefile  
  
298 # EXPORT DELETE END  
  
300 include ../Makefile.targ  
  
302 #  
303 # Cross-reference customization: build a cross-reference over all of the  
304 # sun4u-related directories.  
305 #  
306 SHARED_XRDIRS = ../sun4u ../sun4 ../sfmmu ../sparc ../sun ../common  
307 CLOSED_XRDIRS = $(SHARED_XRDIRS:../%=./% ../../../closed/uts/%)  
308 XRDIRS        = $(SHARED_XRDIRS)  
309 $(CLOSED_BUILD)XRDIRS = $(CLOSED_XRDIRS:../../../closed/uts/sfmmu=)  
  
311 XRPRUNE = i86pc  
  
313 cscope.out tags: FRC  
314     $(XREF) -x $@
```

```

*****
7895 Sun May 26 18:24:07 2013
new/usr/src/uts/sun4u/Makefile.rules
XXX Remove awk(1)
*****
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 # This Makefile defines the build rules for the directory uts/sun4u
27 # and its children. These are the source files which sun4u
28 # "implementation architecture" dependent.
29 #
30 # The following two-level ordering must be maintained in this file.
31 # Lines are sorted first in order of decreasing specificity based on
32 # the first directory component. That is, sun4u rules come before
33 # sparc rules come before common rules.
34 #
35 # Lines whose initial directory components are equal are sorted
36 # alphabetically by the remaining components.
37 #
38 #
39 # Section 1a: C object build rules
40 #
41 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/cpu/%.c
42 $(COMPILE.c) -o $@ $<
43 $(CTFCONVERT_O)
44 #
45 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/cpu/%.s
46 $(COMPILE.s) -o $@ $<
47 #
48 #
49 # This rule for io/ppm/%.c needs to come before the io/%.c rule so that
50 # the sun4u/ppm driver pulls in io/ppm/ppm.c instead of io/ppm.c.
51 # Making an explicit rule to build ppm.o doesn't work because it breaks
52 # the build of excalibur/xcalppm
53 #
54 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/io/ppm/%.c
55 $(COMPILE.c) -o $@ $<
56 $(CTFCONVERT_O)
57 #
58 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ppm/%.c
59 $(COMPILE.c) -o $@ $<
60 $(CTFCONVERT_O)

```

```

62 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/io/%.s
63 $(COMPILE.s) -o $@ $<
64 #
65 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/io/%.c
66 $(COMPILE.c) -o $@ $<
67 $(CTFCONVERT_O)
68 #
69 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/io/i2c/clients/%.c
70 $(COMPILE.c) -o $@ $<
71 $(CTFCONVERT_O)
72 #
73 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/io/i2c/misc/%.c
74 $(COMPILE.c) -o $@ $<
75 $(CTFCONVERT_O)
76 #
77 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/io/i2c/nexus/%.c
78 $(COMPILE.c) -o $@ $<
79 $(CTFCONVERT_O)
80 #
81 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/io/pci/%.c
82 $(COMPILE.c) -o $@ $<
83 $(CTFCONVERT_O)
84 #
85 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/io/pci/%.s
86 $(COMPILE.s) -o $@ $<
87 #
88 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/io/px/%.c
89 $(COMPILE.c) -o $@ $<
90 $(CTFCONVERT_O)
91 #
92 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/io/px/%.s
93 $(COMPILE.s) -o $@ $<
94 #
95 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/io/pciex/%.c
96 $(COMPILE.c) -o $@ $<
97 $(CTFCONVERT_O)
98 #
99 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/io/pciex/%.s
100 $(COMPILE.s) -o $@ $<
101 #
102 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/ml/%.s
103 $(COMPILE.s) -o $@ $<
104 #
105 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/os/%.c
106 $(COMPILE.c) -o $@ $<
107 $(CTFCONVERT_O)
108 #
109 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/pcbe/%.c
110 $(COMPILE.c) -o $@ $<
111 $(CTFCONVERT_O)
112 #
113 $(OBJSDIR)/%.o: $(UTSBASE)/sun4/brand/snl/%.s
114 $(COMPILE.s) -o $@ $<
115 #
116 $(OBJSDIR)/%.o: $(UTSBASE)/sun4/brand/solaris10/%.s
117 $(COMPILE.s) -o $@ $<
118 #
119 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/vm/%.c
120 $(COMPILE.c) -o $@ $<
121 $(CTFCONVERT_O)
122 #
123 $(OBJSDIR)/%.o: $(UTSBASE)/sun4u/vm/%.s
124 $(COMPILE.s) -o $@ $<
125 #
126 $(OBJSDIR)/%.o: $(UTSBASE)/sfmmu/ml/%.s
127 $(COMPILE.s) -o $@ $<

```

```

129 $(OBJSDIR)/%.o: $(UTSBASE)/sfmmu/vm/%.c
130 $(COMPILE.c) -o $@ $<
131 $(CTFCONVERT_O)

133 $(OBJSDIR)/%.o: $(UTSBASE)/sparc/os/%.c
134 $(COMPILE.c) -o $@ $<
135 $(CTFCONVERT_O)

137 $(OBJSDIR)/%.o: $(UTSBASE)/common/os/%.c
138 $(COMPILE.c) -o $@ $<
139 $(CTFCONVERT_O)

141 $(OBJSDIR)/%.o: $(SRC)/common/bignum/sun4u/%.s
142 $(COMPILE.s) -o $@ $<

144 $(OBJSDIR)/%.o: $(SRC)/common/crypto/aes/%.c
145 $(COMPILE.c) -o $@ $<
146 $(CTFCONVERT_O)

148 $(OBJSDIR)/%.o: $(SRC)/common/crypto/arcfour/%.c
149 $(COMPILE.c) -o $@ $<
150 $(CTFCONVERT_O)

152 $(OBJSDIR)/%.o: $(SRC)/common/crypto/des/%.c
153 $(COMPILE.c) -o $@ $<
154 $(CTFCONVERT_O)

156 $(OBJSDIR)/%.o: $(SRC)/common/crypto/aes/sun4u/%.s
157 $(COMPILE.s) -o $@ $<

159 $(OBJSDIR)/%.o: $(SRC)/common/crypto/arcfour/sun4u/%.s
160 $(COMPILE.s) -o $@ $<

162 $(OBJSDIR)/%.o: $(SRC)/common/crypto/des/sun4u/%.s
163 $(COMPILE.s) -o $@ $<

165 $(OBJSDIR)/%.o: $(SRC)/common/crypto/shal/sparc/sun4u/%.s
166 $(COMPILE.s) -o $@ $<

168 $(OBJSDIR)/%.o: $(SRC)/common/atomic/sparcv9/%.s
169 $(COMPILE.s) -o $@ $<

171 $(OBJSDIR)/dtracestubs.s: $(UNIX_O) $(LIBS)
172 $(NM) -u $(UNIX_O) $(LIBS) | $(GREP) __dtrace_probe_ | $(SORT) | \
173 $(UNIQ) | $(AWK) '{ \
173 $(UNIQ) | $(NAWK) '{ \
174 printf("\t.global %s\n\t.type %s, #function\n%s:\n", \
175 $1, $1, $1); }' > $(OBJSDIR)/dtracestubs.s

177 $(DTRACESTUBS): $(DTRACESTUBS_O)
178 $(BUILD.SO) $(DTRACESTUBS_O)

180 #
181 # Section 1b: Lint 'object' build rules
182 #
183 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/cpu/%.c
184 @$(LHEAD) $(LINT.c) $< $(LTAIL))

186 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/cpu/%.s
187 @$(LHEAD) $(LINT.s) $< $(LTAIL))

189 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/io/ppm/%.c
190 @$(LHEAD) $(LINT.c) $< $(LTAIL))

192 $(LINTSDIR)/%.ln: $(UTSBASE)/common/io/ppm/%.c

```

```

193 @$(LHEAD) $(LINT.c) $< $(LTAIL))

195 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/io/%.c
196 @$(LHEAD) $(LINT.c) $< $(LTAIL))

198 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/io/%.s
199 @$(LHEAD) $(LINT.s) $< $(LTAIL))

201 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/io/i2c/clients/%.c
202 @$(LHEAD) $(LINT.c) $< $(LTAIL))

204 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/io/i2c/misc/%.c
205 @$(LHEAD) $(LINT.c) $< $(LTAIL))

207 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/io/i2c/nexus/%.c
208 @$(LHEAD) $(LINT.c) $< $(LTAIL))

210 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/io/pci/%.c
211 @$(LHEAD) $(LINT.c) $< $(LTAIL))

213 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/io/pci/%.s
214 @$(LHEAD) $(LINT.s) $< $(LTAIL))

216 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/io/px/%.c
217 @$(LHEAD) $(LINT.c) $< $(LTAIL))

219 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/io/px/%.s
220 @$(LHEAD) $(LINT.s) $< $(LTAIL))

222 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/io/pciex/%.c
223 @$(LHEAD) $(LINT.c) $< $(LTAIL))

225 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/io/pciex/%.s
226 @$(LHEAD) $(LINT.s) $< $(LTAIL))

228 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/ml/%.s
229 @$(LHEAD) $(LINT.s) $< $(LTAIL))

231 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/os/%.c
232 @$(LHEAD) $(LINT.c) $< $(LTAIL))

234 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/pcbe/%.c
235 @$(LHEAD) $(LINT.c) $< $(LTAIL))

237 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4/brand/snl/%.s
238 @$(LHEAD) $(LINT.s) $< $(LTAIL))

240 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4/brand/solaris10/%.s
241 @$(LHEAD) $(LINT.s) $< $(LTAIL))

243 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/vm/%.c
244 @$(LHEAD) $(LINT.c) $< $(LTAIL))

246 $(LINTSDIR)/%.ln: $(UTSBASE)/sun4u/vm/%.s
247 @$(LHEAD) $(LINT.s) $< $(LTAIL))

249 $(LINTSDIR)/%.ln: $(UTSBASE)/sfmmu/ml/%.s
250 @$(LHEAD) $(LINT.s) $< $(LTAIL))

252 $(LINTSDIR)/%.ln: $(UTSBASE)/sfmmu/vm/%.c
253 @$(LHEAD) $(LINT.c) $< $(LTAIL))

255 $(LINTSDIR)/%.ln: $(UTSBASE)/sparc/os/%.c
256 @$(LHEAD) $(LINT.c) $< $(LTAIL))

258 $(LINTSDIR)/%.ln: $(UTSBASE)/common/os/%.c

```

```
259         @$(LHEAD) $(LINT.c) $< $(LTAIL))
261 $(LINTS_DIR)/%.ln:          $(SRC)/common/bignum/sun4u/%.s
262     @$(LHEAD) $(LINT.s) $< $(LTAIL))
264 $(LINTS_DIR)/%.ln:          $(SRC)/common/crypto/aes/%.c
265     @$(LHEAD) $(LINT.c) $< $(LTAIL))
267 $(LINTS_DIR)/%.ln:          $(SRC)/common/crypto/arcfour/%.c
268     @$(LHEAD) $(LINT.c) $< $(LTAIL))
270 $(LINTS_DIR)/%.ln:          $(SRC)/common/crypto/des/%.c
271     @$(LHEAD) $(LINT.c) $< $(LTAIL))
273 $(LINTS_DIR)/%.ln:          $(SRC)/common/crypto/aes/sun4u/%.s
274     @$(LHEAD) $(LINT.s) $< $(LTAIL))
276 $(LINTS_DIR)/%.ln:          $(SRC)/common/crypto/arcfour/sun4u/%.s
277     @$(LHEAD) $(LINT.s) $< $(LTAIL))
279 $(LINTS_DIR)/%.ln:          $(SRC)/common/crypto/des/sun4u/%.s
280     @$(LHEAD) $(LINT.s) $< $(LTAIL))
282 $(LINTS_DIR)/%.ln:          $(SRC)/common/crypto/shal/sparc/sun4u/%.s
283     @$(LHEAD) $(LINT.s) $< $(LTAIL))
285 $(LINTS_DIR)/%.ln:          $(SRC)/common/atomic/%.c
286     @$(LHEAD) $(LINT.c) $< $(LTAIL))
```

```

*****
7397 Sun May 26 18:24:07 2013
new/usr/src/uts/sun4v/Makefile.rules
XXX Remove awk(1)
*****
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) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 #
26 #
27 # This Makefile defines the build rules for the directory uts/sun4v
28 # and its children. These are the source files which sun4v
29 # "implementation architecture" dependent.
30 #
31 # The following two-level ordering must be maintained in this file.
32 # Lines are sorted first in order of decreasing specificity based on
33 # the first directory component. That is, sun4v rules come before
34 # sparc rules come before common rules.
35 #
36 # Lines whose initial directory components are equal are sorted
37 # alphabetically by the remaining components.
38 #
39 #
40 # Section 1a: C object build rules
41 #
42 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/cpu/%.c
43 $(COMPILE.c) -o $@ $<
44 $(CTFCONVERT_O)
45 #
46 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/cpu/%.s
47 $(COMPILE.s) -o $@ $<
48 #
49 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/io/%.c
50 $(COMPILE.c) -o $@ $<
51 $(CTFCONVERT_O)
52 #
53 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/ml/%.s
54 $(COMPILE.s) -o $@ $<
55 #
56 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/os/%.c
57 $(COMPILE.c) -o $@ $<
58 $(CTFCONVERT_O)
59 #
60 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/pcbe/%.c
61 $(COMPILE.c) -o $@ $<

```

```

62 $(CTFCONVERT_O)
63 #
64 $(OBJSDIR)/%.o: $(UTSBASE)/sun4/brand/snl/%.s
65 $(COMPILE.s) -o $@ $<
66 #
67 $(OBJSDIR)/%.o: $(UTSBASE)/sun4/brand/solaris10/%.s
68 $(COMPILE.s) -o $@ $<
69 #
70 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/promif/%.c
71 $(COMPILE.c) -o $@ $<
72 $(CTFCONVERT_O)
73 #
74 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/io/px/%.c
75 $(COMPILE.c) -o $@ $<
76 $(CTFCONVERT_O)
77 #
78 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/io/px/%.s
79 $(COMPILE.s) -o $@ $<
80 #
81 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/io/fpc/%.c
82 $(COMPILE.c) -o $@ $<
83 $(CTFCONVERT_O)
84 #
85 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/io/fpc/%.s
86 $(COMPILE.s) -o $@ $<
87 #
88 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/io/n2piupc/%.c
89 $(COMPILE.c) -o $@ $<
90 $(CTFCONVERT_O)
91 #
92 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/io/n2piupc/%.s
93 $(COMPILE.s) -o $@ $<
94 #
95 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/io/iospc/%.c
96 $(COMPILE.c) -o $@ $<
97 $(CTFCONVERT_O)
98 #
99 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/io/iospc/%.s
100 $(COMPILE.s) -o $@ $<
101 #
102 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/io/glvc/%.c
103 $(COMPILE.c) -o $@ $<
104 $(CTFCONVERT_O)
105 #
106 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/io/glvc/%.s
107 $(COMPILE.s) -o $@ $<
108 #
109 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/io/pciex/%.c
110 $(COMPILE.c) -o $@ $<
111 $(CTFCONVERT_O)
112 #
113 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/io/pciex/%.s
114 $(COMPILE.s) -o $@ $<
115 #
116 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/vm/%.c
117 $(COMPILE.c) -o $@ $<
118 $(CTFCONVERT_O)
119 #
120 $(OBJSDIR)/%.o: $(UTSBASE)/sun4v/vm/%.s
121 $(COMPILE.s) -o $@ $<
122 #
123 $(OBJSDIR)/%.o: $(UTSBASE)/sfmmu/ml/%.s
124 $(COMPILE.s) -o $@ $<
125 #
126 $(OBJSDIR)/%.o: $(UTSBASE)/sfmmu/vm/%.c
127 $(COMPILE.c) -o $@ $<

```

```

128     $(CTFCONVERT_O)

130 $(OBJS_DIR)/%.o:                $(UTSBASE)/sparc/os/%.c
131     $(COMPILE.c) -o $@ $<
132     $(CTFCONVERT_O)

134 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/os/%.c
135     $(COMPILE.c) -o $@ $<
136     $(CTFCONVERT_O)

138 $(OBJS_DIR)/%.o:                $(SRC)/common/mdesc/%.c
139     $(COMPILE.c) -o $@ $<
140     $(CTFCONVERT_O)

142 $(OBJS_DIR)/%.o:                $(UTSBASE)/sun4v/io/niux/%.c
143     $(COMPILE.c) -o $@ $<
144     $(CTFCONVERT_O)

146 $(OBJS_DIR)/%.o:                $(UTSBASE)/sun4v/io/n2rng/%.c
147     $(COMPILE.c) -o $@ $<
148     $(CTFCONVERT_O)

150 $(OBJS_DIR)/%.o:                $(UTSBASE)/sun4v/io/%.s
151     $(COMPILE.s) -o $@ $<

153 $(OBJS_DIR)/%.o:                $(UTSBASE)/sun4v/io/n2rng/%.s
154     $(COMPILE.s) -o $@ $<

156 $(OBJS_DIR)/%.o:                $(SRC)/common/atomic/sparcv9/%.s
157     $(COMPILE.s) -o $@ $<

159 $(OBJS_DIR)/%.o:                $(SRC)/common/crypto/arcfour/sun4v/%.c
160     $(COMPILE.c) -o $@ $<
161     $(CTFCONVERT_O)

163 $(OBJS_DIR)/dtracestubs.s:      $(UNIX_O) $(LIBS)
164     $(NM) -u $(UNIX_O) $(LIBS) | $(GREP) __dtrace_probe_ | $(SORT) | \
165     $(UNIQ) | $(AWK) '{ \
166     $(UNIQ) } $(NAWK) '{ \
166     printf("\t.global %s\n\t.type %s, #function\n%s:\n", \
167     $$1, $$1, $$1); }' > $(OBJS_DIR)/dtracestubs.s

169 $(DTRACESTUBS): $(DTRACESTUBS_O)
170     $(BUILD.SO) $(DTRACESTUBS_O)

172 #
173 # Section lb: Lint 'object' build rules
174 #
175 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/cpu/%.c
176     @$(LHEAD) $(LINT.c) $< $(LTAIL))

178 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/cpu/%.s
179     @$(LHEAD) $(LINT.s) $< $(LTAIL))

181 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/%.c
182     @$(LHEAD) $(LINT.c) $< $(LTAIL))

184 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/%.s
185     @$(LHEAD) $(LINT.s) $< $(LTAIL))

187 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/px/%.c
188     @$(LHEAD) $(LINT.c) $< $(LTAIL))

190 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/px/%.s
191     @$(LHEAD) $(LINT.s) $< $(LTAIL))

```

```

193 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/niux/%.c
194     @$(LHEAD) $(LINT.c) $< $(LTAIL))

196 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/fpc/%.c
197     @$(LHEAD) $(LINT.c) $< $(LTAIL))

199 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/fpc/%.s
200     @$(LHEAD) $(LINT.s) $< $(LTAIL))

202 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/n2piupc/%.c
203     @$(LHEAD) $(LINT.c) $< $(LTAIL))

205 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/n2piupc/%.s
206     @$(LHEAD) $(LINT.s) $< $(LTAIL))

208 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/iospc/%.c
209     @$(LHEAD) $(LINT.c) $< $(LTAIL))

211 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/iospc/%.s
212     @$(LHEAD) $(LINT.s) $< $(LTAIL))

214 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/glvc/%.c
215     @$(LHEAD) $(LINT.c) $< $(LTAIL))

217 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/glvc/%.s
218     @$(LHEAD) $(LINT.s) $< $(LTAIL))

220 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/n2rng/%.c
221     @$(LHEAD) $(LINT.c) $< $(LTAIL))

223 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/n2rng/%.s
224     @$(LHEAD) $(LINT.s) $< $(LTAIL))

226 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/pciex/%.c
227     @$(LHEAD) $(LINT.c) $< $(LTAIL))

229 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/io/pciex/%.s
230     @$(LHEAD) $(LINT.s) $< $(LTAIL))

232 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/ml/%.s
233     @$(LHEAD) $(LINT.s) $< $(LTAIL))

235 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/os/%.c
236     @$(LHEAD) $(LINT.c) $< $(LTAIL))

238 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/pcbe/%.c
239     @$(LHEAD) $(LINT.c) $< $(LTAIL))

241 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4/brand/snl/%.s
242     @$(LHEAD) $(LINT.s) $< $(LTAIL))

244 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4/brand/solaris10/%.s
245     @$(LHEAD) $(LINT.s) $< $(LTAIL))

247 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/promif/%.c
248     @$(LHEAD) $(LINT.c) $< $(LTAIL))

250 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/vm/%.c
251     @$(LHEAD) $(LINT.c) $< $(LTAIL))

253 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sun4v/vm/%.s
254     @$(LHEAD) $(LINT.s) $< $(LTAIL))

256 $(LINTS_DIR)/%.ln:                $(UTSBASE)/sfmmu/ml/%.s
257     @$(LHEAD) $(LINT.s) $< $(LTAIL))

```

new/usr/src/uts/sun4v/Makefile.rules

5

```
259 $(LINTSDIR)/%.ln: $(UTSBASE)/sfmmu/vm/%.c
260     @$(LHEAD) $(LINT.c) $< $(LTAIL))

262 $(LINTSDIR)/%.ln: $(UTSBASE)/sparc/os/%.c
263     @$(LHEAD) $(LINT.c) $< $(LTAIL))

265 $(LINTSDIR)/%.ln: $(UTSBASE)/common/os/%.c
266     @$(LHEAD) $(LINT.c) $< $(LTAIL))

268 $(LINTSDIR)/%.ln: $(SRC)/common/mdesc/%.c
269     @$(LHEAD) $(LINT.c) $< $(LTAIL))

271 $(LINTSDIR)/%.ln: $(SRC)/common/atomic/%.c
272     @$(LHEAD) $(LINT.c) $< $(LTAIL))

274 $(LINTSDIR)/%.ln: $(SRC)/common/crypto/arcfour/sun4v/%.c
275     @$(LHEAD) $(LINT.c) $< $(LTAIL))
```