

```

*****
70075 Mon Sep  9 17:55:47 2013
new/usr/src/Targetdirs
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
1 # CDDL HEADER START
2 #
3 # The contents of this file are subject to the terms of the
4 # Common Development and Distribution License (the "License").
5 # You may not use this file except in compliance with the License.
6 #
7 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
8 # or http://www.opensolaris.org/os/licensing.
9 # See the License for the specific language governing permissions
10 # and limitations under the License.
11 #
12 # When distributing Covered Code, include this CDDL HEADER in each
13 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
14 # If applicable, add the following below this CDDL HEADER, with the
15 # fields enclosed by brackets "[]" replaced with your own identifying
16 # information: Portions Copyright [yyyy] [name of copyright owner]
17 #
18 # CDDL HEADER END
19 #
20 #
21 #
22 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright 2011, Richard Lowe
24 # Copyright 2011 Nexenta Systems, Inc. All rights reserved.
25 # Copyright (c) 2012 by Delphix. All rights reserved.
26 # Copyright 2012 OmniTI Computer Consulting, Inc. All rights reserved.
27 # Copyright (c) 2013 RackTop Systems.
28 #endif /* ! codereview */
29 #
30 #
31 #
32 # It is easier to think in terms of directory names without the ROOT macro
33 # prefix.  ROOTDIRS is TARGETDIRS with ROOT prefixes.  It is necessary
34 # to work with ROOT prefixes when controlling conditional assignments.
35 #
36 #
37 DIRLINKS=      $(SYM.DIRS)
38 $(BUILD64)    DIRLINKS += $(SYM.DIRS64)
39 #
40 FILELINKS= $(SYM.USRCCSLIB) $(SYM.USRLIB)
41 $(BUILD64)   FILELINKS += $(SYM.USRCCSLIB64) $(SYM.USRLIB64)
42 #
43 TARGETDIRS=   $(DIRS)
44 $(BUILD64)   TARGETDIRS += $(DIRS64)
45 #
46 TARGETDIRS   += $(FILELINKS) $(DIRLINKS)
47 #
48 i386_DIRS=    \
49 /boot/acpi    \
50 /boot/acpi/tables \
51 /boot/grub    \
52 /boot/grub/bin \
53 /platform/i86pc \
54 /usr/lib/xen  \
55 /usr/lib/xen/bin \
56 #
57 sparc_DIRS=   \

```

```

58 /usr/lib/ldoms
59 #
60 sparc_64ONLY= $(POUND_SIGN)
61 64ONLY=      $(MACH)_64ONLY)
62 #
63 $(64ONLY) MACH32_DIRS=/usr/ucb/$(MACH32)
64 #
65 DIRS= \
66 /boot \
67 /boot/solaris \
68 /boot/solaris/bin \
69 $(MACH)_DIRS \
70 /dev \
71 /dev/dsk \
72 /dev/fd \
73 /dev/ipnet \
74 /dev/net \
75 /dev/rdisk \
76 /dev/rmt \
77 /dev/pts \
78 /dev/sad \
79 /dev/swap \
80 /dev/term \
81 /dev/vt \
82 /dev/zcons \
83 /devices \
84 /devices/pseudo \
85 /etc \
86 /etc/brand \
87 /etc/brand/solaris10 \
88 /etc/cron.d \
89 /etc/crypto \
90 /etc/crypto/certs \
91 /etc/crypto/crls \
92 /etc/dbus-1 \
93 /etc/dbus-1/system.d \
94 /etc/default \
95 /etc/devices \
96 /etc/dev \
97 /etc/dfs \
98 /etc/dladm \
99 /etc/fs \
100 /etc/fs/nfs \
101 /etc/fs/zfs \
102 /etc/ftpd \
103 /etc/hal \
104 /etc/hal/fdi \
105 /etc/hal/fdi/information \
106 /etc/hal/fdi/information/10freedesktop \
107 /etc/hal/fdi/information/20thirdparty \
108 /etc/hal/fdi/information/30user \
109 /etc/hal/fdi/policy \
110 /etc/hal/fdi/policy/10osvendor \
111 /etc/hal/fdi/policy/20thirdparty \
112 /etc/hal/fdi/policy/30user \
113 /etc/hal/fdi/preprobe \
114 /etc/hal/fdi/preprobe/10osvendor \
115 /etc/hal/fdi/preprobe/20thirdparty \
116 /etc/hal/fdi/preprobe/30user \
117 /etc/ipadm \
118 /etc/iscsi \
119 /etc/rpcsec \
120 /etc/security \
121 /etc/security/auth_attr.d \
122 /etc/security/exec_attr.d \
123 /etc/security/prof_attr.d \

```

```

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

```

```

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

```

```

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

```

```

322 /usr/share/gnome \
323 /usr/share/gnome/autostart \
324 /usr/share/hwdata \
325 /usr/share/lib \
326 /usr/share/lib/ccs \
327 /usr/share/lib/tmac \
328 /usr/share/lib/ldif \
329 /usr/share/lib/xml \
330 /usr/share/lib/xml/dtd \
331 /usr/share/man \
332 /usr/share/man/man1 \
333 /usr/share/man/man1b \
334 /usr/share/man/man1c \
335 /usr/share/man/man1m \
336 /usr/share/man/man2 \
337 /usr/share/man/man3 \
338 /usr/share/man/man3bsm \
339 /usr/share/man/man3c \
340 /usr/share/man/man3c_db \
341 /usr/share/man/man3cfgadm \
342 /usr/share/man/man3commputil \
343 /usr/share/man/man3contract \
344 /usr/share/man/man3opc \
345 /usr/share/man/man3curses \
346 /usr/share/man/man3dat \
347 /usr/share/man/man3devind \
348 /usr/share/man/man3devinfo \
349 /usr/share/man/man3dlpi \
350 /usr/share/man/man3dns_sd \
351 /usr/share/man/man3elf \
352 /usr/share/man/man3execct \
353 /usr/share/man/man3ext \
354 /usr/share/man/man3fcoe \
355 /usr/share/man/man3fstyp \
356 /usr/share/man/man3gen \
357 /usr/share/man/man3gss \
358 /usr/share/man/man3head \
359 /usr/share/man/man3iscsit \
360 /usr/share/man/man3kstat \
361 /usr/share/man/man3kvm \
362 /usr/share/man/man3ldap \
363 /usr/share/man/man3lgrp \
364 /usr/share/man/man3lib \
365 /usr/share/man/man3libucb \
366 /usr/share/man/man3mail \
367 /usr/share/man/man3malloc \
368 /usr/share/man/man3mp \
369 /usr/share/man/man3mpapi \
370 /usr/share/man/man3nsl \
371 /usr/share/man/man3nvpair \
372 /usr/share/man/man3pam \
373 /usr/share/man/man3papi \
374 /usr/share/man/man3perl \
375 /usr/share/man/man3picl \
376 /usr/share/man/man3picltree \
377 /usr/share/man/man3pool \
378 /usr/share/man/man3proc \
379 /usr/share/man/man3project \
380 /usr/share/man/man3resolv \
381 /usr/share/man/man3rpc \
382 /usr/share/man/man3rsm \
383 /usr/share/man/man3sas1 \
384 /usr/share/man/man3scf \
385 /usr/share/man/man3sec \
386 /usr/share/man/man3secdb \
387 /usr/share/man/man3sip \

```

```

388 /usr/share/man/man3slp \
389 /usr/share/man/man3socket \
390 /usr/share/man/man3stmf \
391 /usr/share/man/man3sysevent \
392 /usr/share/man/man3tecla \
393 /usr/share/man/man3tnf \
394 /usr/share/man/man3tsol \
395 /usr/share/man/man3ucb \
396 /usr/share/man/man3uuid \
397 /usr/share/man/man3volmgt \
398 /usr/share/man/man3xcurses \
399 /usr/share/man/man3xnet \
400 /usr/share/man/man4 \
401 /usr/share/man/man5 \
402 /usr/share/man/man7 \
403 /usr/share/man/man7d \
404 /usr/share/man/man7fs \
405 /usr/share/man/man7i \
406 /usr/share/man/man7ipp \
407 /usr/share/man/man7m \
408 /usr/share/man/man7p \
409 /usr/share/man/man9 \
410 /usr/share/man/man9e \
411 /usr/share/man/man9f \
412 /usr/share/man/man9p \
413 /usr/share/man/man9s \
414 /usr/share/src \
415 /usr/snadm \
416 /usr/snadm/lib \
417 /usr/ucb \
418 $(MACH32_DIRS) \
419 /usr/ucb/lib \
420 /usr/xpg4 \
421 /usr/xpg4/bin \
422 /usr/xpg4/include \
423 /usr/xpg4/lib \
424 /usr/xpg6 \
425 /usr/xpg6/bin \
426 /var \
427 /var/adm \
428 /var/adm/exacct \
429 /var/adm/log \
430 /var/adm/pool \
431 /var/adm/sa \
432 /var/adm/sm.bin \
433 /var/adm/streams \
434 /var/cores \
435 /var/cron \
436 /var/db \
437 /var/db/ipf \
438 /var/games \
439 /var/idmap \
440 /var/krb5 \
441 /var/krb5/rcache \
442 /var/krb5/rcache/root \
443 /var/ld \
444 /var/log \
445 /var/log/pool \
446 /var/logadm \
447 /var/mail \
448 /var/news \
449 /var/opt \
450 /var/preserve \
451 /var/run \
452 /var/saf \
453 /var/sadm \

```

```

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

507 sparcv9_DIRS64= \
508 /platform/sun4u \
509 /platform/sun4u/lib \
510 /platform/sun4u/lib/$(MACH64) \
511 /usr/platform/sun4u \
512 /usr/platform/sun4u/sbin \
513 /usr/platform/sun4u/lib \
514 /platform/sun4v/lib \
515 /platform/sun4v/lib/$(MACH64) \
516 /usr/platform/sun4v/sbin \
517 /usr/platform/sun4v/lib \
518 /usr/platform/sun4u-us3/lib \
519 /usr/platform/sun4u-opl/lib

```

```

521 amd64_DIRS64= \
522     /platform/i86pc/amd64

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

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

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

```

```

586     /usr/lib/libp/32 \
587     /usr/lib/lwp/32 \
588     /usr/lib/link_audit/32 \
589     /usr/lib/secure/32 \
590     /usr/mail \
591     /usr/man \
592     /usr/pub \
593     /usr/src \
594     /usr/tmp \
595     /usr/ucblib/32 \
596     /var/ld/32

598 sparc_SYM.DIRS64=

600 SYM.DIRS64= \
601     ${MACH}_SYM.DIRS64 \
602     /lib/64 \
603     /lib/crypto/64 \
604     /lib/secure/64 \
605     /usr/lib/64 \
606     /usr/lib/brand/sn1/64 \
607     /usr/lib/brand/solaris10/64 \
608     /usr/lib/elfedit/64 \
609     /usr/lib/libp/64 \
610     /usr/lib/link_audit/64 \
611     /usr/lib/lwp/64 \
612     /usr/lib/secure/64 \
613     /usr/lib/security/64 \
614     /usr/xpg4/lib/64 \
615     /var/ld/64 \
616     /usr/ucblib/64

618 # prepend the ROOT prefix

620 ROOTDIRS=      $(TARGETDIRS:%=${ROOT}%)

622 # conditional assignments
623 #
624 # Target directories with non-default values for owner and group must
625 # be referenced here, using their fully-prefixed names, and the non-
626 # default values assigned. If a directory is mentioned above and not
627 # mentioned below, it has default values for attributes.
628 #
629 # The default value for DIRMODE is specified in usr/src/Makefile.master.
630 #

632 $(ROOT)/var/adm \
633 $(ROOT)/var/adm/sa :=          DIRMODE= 775

635 $(ROOT)/var/spool/lp:=        DIRMODE= 775

637 # file mode
638 #
639 $(ROOT)/tmp \
640 $(ROOT)/var/krb5/rcache \
641 $(ROOT)/var/preserve \
642 $(ROOT)/var/spool/pkg \
643 $(ROOT)/var/spool/uucppublic \
644 $(ROOT)/var/tmp:=            DIRMODE= 1777

646 $(ROOT)/root:=              DIRMODE= 700

648 $(ROOT)/var/krb5/rcache/root:= DIRMODE= 700

651 #

```

```

652 # These permissions must match those set
653 # in the package manifests.
654 #
655 $(ROOT)/var/sadm/pkg \
656 $(ROOT)/var/sadm/security \
657 $(ROOT)/var/sadm/install/logs :=          DIRMODE= 555

660 #
661 # These permissions must match the ones set
662 # internally by fdfs and autofs.
663 #
664 $(ROOT)/dev/fd \
665 $(ROOT)/home:=                          DIRMODE= 555

667 $(ROOT)/var/mail:=                      DIRMODE=1777

669 $(ROOT)/proc:=                          DIRMODE= 555

671 $(ROOT)/system/contract:=               DIRMODE= 555
672 $(ROOT)/system/object:=                 DIRMODE= 555

674 # symlink assignments, LINKDEST is the value of the symlink
675 #
676 $(ROOT)/usr/lib/cron:=                   LINKDEST=../etc/cron.d
677 $(ROOT)/bin:=                            LINKDEST=usr/bin
678 $(ROOT)/lib/32:=                         LINKDEST=.
679 $(ROOT)/lib/crypto/32:=                 LINKDEST=.
680 $(ROOT)/lib/secure/32:=                 LINKDEST=.
681 $(ROOT)/dev/stdin:=                     LINKDEST=fd/0
682 $(ROOT)/dev/stdout:=                    LINKDEST=fd/1
683 $(ROOT)/dev/stderr:=                    LINKDEST=fd/2
684 $(ROOT)/usr/pub:=                        LINKDEST=share/lib/pub
685 $(ROOT)/usr/man:=                        LINKDEST=share/man
686 $(ROOT)/usr/src:=                        LINKDEST=share/src
687 $(ROOT)/usr/adm:=                        LINKDEST=../var/adm
688 $(ROOT)/etc/lib/ld.so.1:=                LINKDEST=../lib/ld.so.1
689 $(ROOT)/etc/lib/libdl.so.1:=             LINKDEST=../lib/libdl.so.1
690 $(ROOT)/etc/lib/nss_files.so.1:=         LINKDEST=../lib/nss_files.so.1
691 $(ROOT)/etc/log:=                        LINKDEST=../var/adm/log
692 $(ROOT)/usr/mail:=                       LINKDEST=../var/mail
693 $(ROOT)/usr/news:=                       LINKDEST=../var/news
694 $(ROOT)/usr/preserve:=                   LINKDEST=../var/preserve
695 $(ROOT)/usr/spool:=                      LINKDEST=../var/spool
696 $(ROOT)/usr/tmp:=                        LINKDEST=../var/tmp
697 $(ROOT)/usr/lib/tmac:=                   LINKDEST=../share/lib/tmac
698 $(ROOT)/usr/lib/32:=                     LINKDEST=.
699 $(ROOT)/usr/lib/elfedit/32:=             LINKDEST=.
700 $(ROOT)/usr/lib/libp/32:=                 LINKDEST=.
701 $(ROOT)/usr/lib/lwp/32:=                 LINKDEST=.
702 $(ROOT)/usr/lib/link_audit/32:=          LINKDEST=.
703 $(ROOT)/usr/lib/secure/32:=              LINKDEST=.
704 $(ROOT)/usr/ccs/lib/link_audit:=         LINKDEST=../lib/link_audit
705 $(ROOT)/var/ld/32:=                      LINKDEST=.
706 $(ROOT)/usr/ucblib/32:=                  LINKDEST=.

709 $(BUILD64) $(ROOT)/lib/64:=              LINKDEST=$(MACH64)
710 $(BUILD64) $(ROOT)/lib/crypto/64:=       LINKDEST=$(MACH64)
711 $(BUILD64) $(ROOT)/lib/secure/64:=       LINKDEST=$(MACH64)
712 $(BUILD64) $(ROOT)/usr/lib/64:=          LINKDEST=$(MACH64)
713 $(BUILD64) $(ROOT)/usr/lib/elfedit/64:=   LINKDEST=$(MACH64)
714 $(BUILD64) $(ROOT)/usr/lib/brand/sn1/64:= LINKDEST=$(MACH64)
715 $(BUILD64) $(ROOT)/usr/lib/brand/solaris10/64:= LINKDEST=$(MACH64)
716 $(BUILD64) $(ROOT)/usr/lib/libp/64:=     LINKDEST=$(MACH64)
717 $(BUILD64) $(ROOT)/usr/lib/lwp/64:=       LINKDEST=$(MACH64)

```

```

718 $(BUILD64) $(ROOT)/usr/lib/link_audit/64:= LINKDEST=$(MACH64)
719 $(BUILD64) $(ROOT)/usr/lib/secure/64:=     LINKDEST=$(MACH64)
720 $(BUILD64) $(ROOT)/usr/lib/security/64:=   LINKDEST=$(MACH64)
721 $(BUILD64) $(ROOT)/usr/xpg4/lib/64:=       LINKDEST=$(MACH64)
722 $(BUILD64) $(ROOT)/var/ld/64:=             LINKDEST=$(MACH64)
723 $(BUILD64) $(ROOT)/usr/ucblib/64:=         LINKDEST=$(MACH64)

725 #
726 # Installing a directory symlink calls for overriding INS.dir to install
727 # a symlink.
728 #
729 $(DIRLINKS:%=$(ROOT)%):= \
730     INS.dir= -$(RM) -r $@; $(SYMLINK) $(LINKDEST) $@

732 # Special symlinks to populate usr/ccs/lib, whose objects
733 # have actually been moved to usr/lib
734 # Rather than adding another set of rules, we add usr/lib/lwp files here
735 $(ROOT)/usr/ccs/lib/libcurses.so:=         REALPATH=../lib/libcurses.so.1
736 $(ROOT)/usr/ccs/lib/libl-lcurses:=         REALPATH=../lib/libl-lcurses
737 $(ROOT)/usr/ccs/lib/libl-lcurses.ln:=      REALPATH=../lib/libl-lcurses.ln
738 $(ROOT)/usr/ccs/lib/libform.so:=           REALPATH=../lib/libform.so.1
739 $(ROOT)/usr/ccs/lib/libl-lform:=           REALPATH=../lib/libl-lform
740 $(ROOT)/usr/ccs/lib/libl-lform.ln:=        REALPATH=../lib/libl-lform.ln
741 $(ROOT)/usr/ccs/lib/libgen.so:=            REALPATH=../lib/libgen.so.1
742 $(ROOT)/usr/ccs/lib/libl-lgen:=            REALPATH=../lib/libl-lgen
743 $(ROOT)/usr/ccs/lib/libl-lgen.ln:=         REALPATH=../lib/libl-lgen.ln
744 $(ROOT)/usr/ccs/lib/libmalloc.so:=        REALPATH=../lib/libmalloc.so.1
745 $(ROOT)/usr/ccs/lib/libmenu.so:=          REALPATH=../lib/libmenu.so.1
746 $(ROOT)/usr/ccs/lib/libl-lmenu:=          REALPATH=../lib/libl-lmenu
747 $(ROOT)/usr/ccs/lib/libl-lmenu.ln:=       REALPATH=../lib/libl-lmenu.ln
748 $(ROOT)/usr/ccs/lib/libpanel.so:=          REALPATH=../lib/libpanel.so.1
749 $(ROOT)/usr/ccs/lib/libl-lpanel:=          REALPATH=../lib/libl-lpanel
750 $(ROOT)/usr/ccs/lib/libl-lpanel.ln:=      REALPATH=../lib/libl-lpanel.ln
751 $(ROOT)/usr/ccs/lib/libtermcap.so:=        REALPATH=../lib/libtermcap.so.1
752 $(ROOT)/usr/ccs/lib/libl-ltermcap:=       REALPATH=../lib/libl-lcurses
753 $(ROOT)/usr/ccs/lib/libl-ltermcap.ln:=    REALPATH=../lib/libl-lcurses.ln
754 $(ROOT)/usr/ccs/lib/libl-ltermcap.so:=    REALPATH=../lib/libl-ltermcap.so.1
755 $(ROOT)/usr/ccs/lib/libl-ltermcap.ln:=    REALPATH=../lib/libl-ltermcap
756 $(ROOT)/usr/ccs/lib/libl-ltermcap.ln:=    REALPATH=../lib/libl-ltermcap.ln
757 $(ROOT)/usr/ccs/lib/values-Xa.o:=         REALPATH=../lib/values-Xa.o
758 $(ROOT)/usr/ccs/lib/values-Xc.o:=         REALPATH=../lib/values-Xc.o
759 $(ROOT)/usr/ccs/lib/values-Xs.o:=         REALPATH=../lib/values-Xs.o
760 $(ROOT)/usr/ccs/lib/values-Xt.o:=         REALPATH=../lib/values-Xt.o
761 $(ROOT)/usr/ccs/lib/values-xpg4.o:=       REALPATH=../lib/values-xpg4.o
762 $(ROOT)/usr/ccs/lib/values-xpg6.o:=       REALPATH=../lib/values-xpg6.o
763 $(ROOT)/usr/ccs/lib/libl.so:=              REALPATH=../lib/libl.so.1
764 $(ROOT)/usr/ccs/lib/libl-ll.ln:=          REALPATH=../lib/libl-ll.ln
765 $(ROOT)/usr/ccs/lib/libl-liby.so:=        REALPATH=../lib/liby.so.1
766 $(ROOT)/usr/ccs/lib/libl-ly.ln:=          REALPATH=../lib/libl-ly.ln
767 $(ROOT)/usr/lib/libp/libc.so.1:=          REALPATH=../lib/libc.so.1
768 $(ROOT)/usr/lib/lwp/libthread.so.1:=      REALPATH=../libthread.so.1
769 $(ROOT)/usr/lib/lwp/libthread_db.so.1:=    REALPATH=../libthread_db.so.1

771 # symlinks to populate usr/ccs/lib/$(MACH64)
772 $(ROOT)/usr/ccs/lib/$(MACH64)/libcurses.so:= \
773     REALPATH=../lib/$(MACH64)/libcurses.so.1
774 $(ROOT)/usr/ccs/lib/$(MACH64)/libl-lcurses.ln:= \
775     REALPATH=../lib/$(MACH64)/libl-lcurses.ln
776 $(ROOT)/usr/ccs/lib/$(MACH64)/libform.so:= \
777     REALPATH=../lib/$(MACH64)/libform.so.1
778 $(ROOT)/usr/ccs/lib/$(MACH64)/libl-lform.ln:= \
779     REALPATH=../lib/$(MACH64)/libl-lform.ln
780 $(ROOT)/usr/ccs/lib/$(MACH64)/libgen.so:= \
781     REALPATH=../lib/$(MACH64)/libgen.so.1
782 $(ROOT)/usr/ccs/lib/$(MACH64)/libl-lgen.ln:= \
783     REALPATH=../lib/$(MACH64)/libl-lgen.ln

```

```

784 $(ROOT)/usr/ccs/lib/$(MACH64)/libmalloc.so:= \
785     REALPATH=../../../../lib/$(MACH64)/libmalloc.so.1
786 $(ROOT)/usr/ccs/lib/$(MACH64)/libmenu.so:= \
787     REALPATH=../../../../lib/$(MACH64)/libmenu.so.1
788 $(ROOT)/usr/ccs/lib/$(MACH64)/llib-lmenu.ln:= \
789     REALPATH=../../../../lib/$(MACH64)/llib-lmenu.ln
790 $(ROOT)/usr/ccs/lib/$(MACH64)/libpanel.so:= \
791     REALPATH=../../../../lib/$(MACH64)/libpanel.so.1
792 $(ROOT)/usr/ccs/lib/$(MACH64)/llib-lpanel.ln:= \
793     REALPATH=../../../../lib/$(MACH64)/llib-lpanel.ln
794 $(ROOT)/usr/ccs/lib/$(MACH64)/libtermli.so:= \
795     REALPATH=../../../../lib/$(MACH64)/libcurses.so.1
796 $(ROOT)/usr/ccs/lib/$(MACH64)/llib-ltermli.ln:= \
797     REALPATH=../../../../lib/$(MACH64)/llib-lcurses.ln
798 $(ROOT)/usr/ccs/lib/$(MACH64)/libtermcap.so:= \
799     REALPATH=../../../../lib/$(MACH64)/libtermcap.so.1
800 $(ROOT)/usr/ccs/lib/$(MACH64)/llib-ltermcap.ln:= \
801     REALPATH=../../../../lib/$(MACH64)/llib-ltermcap.ln
802 $(ROOT)/usr/ccs/lib/$(MACH64)/values-Xa.o:= \
803     REALPATH=../../../../lib/$(MACH64)/values-Xa.o
804 $(ROOT)/usr/ccs/lib/$(MACH64)/values-Xc.o:= \
805     REALPATH=../../../../lib/$(MACH64)/values-Xc.o
806 $(ROOT)/usr/ccs/lib/$(MACH64)/values-Xs.o:= \
807     REALPATH=../../../../lib/$(MACH64)/values-Xs.o
808 $(ROOT)/usr/ccs/lib/$(MACH64)/values-Xt.o:= \
809     REALPATH=../../../../lib/$(MACH64)/values-Xt.o
810 $(ROOT)/usr/ccs/lib/$(MACH64)/values-xpg4.o:= \
811     REALPATH=../../../../lib/$(MACH64)/values-xpg4.o
812 $(ROOT)/usr/ccs/lib/$(MACH64)/values-xpg6.o:= \
813     REALPATH=../../../../lib/$(MACH64)/values-xpg6.o
814 $(ROOT)/usr/ccs/lib/$(MACH64)/libl.so:= \
815     REALPATH=../../../../lib/$(MACH64)/libl.so.1
816 $(ROOT)/usr/ccs/lib/$(MACH64)/llib-ll.ln:= \
817     REALPATH=../../../../lib/$(MACH64)/llib-ll.ln
818 $(ROOT)/usr/ccs/lib/$(MACH64)/liby.so:= \
819     REALPATH=../../../../lib/$(MACH64)/liby.so.1
820 $(ROOT)/usr/ccs/lib/$(MACH64)/llib-ly.ln:= \
821     REALPATH=../../../../lib/$(MACH64)/llib-ly.ln
822 $(ROOT)/usr/lib/libp/$(MACH64)/libc.so.1:= \
823     REALPATH=../../../../lib/$(MACH64)/libc.so.1
824 $(ROOT)/usr/lib/lwp/$(MACH64)/libthread.so.1:= \
825     REALPATH=../../../../$(MACH64)/libthread.so.1
826 $(ROOT)/usr/lib/lwp/$(MACH64)/libthread_db.so.1:= \
827     REALPATH=../../../../$(MACH64)/libthread_db.so.1

829 SYM.USRCCSLIB= \
830     /usr/ccs/lib/libcurses.so \
831     /usr/ccs/lib/llib-lcurses \
832     /usr/ccs/lib/llib-lcurses.ln \
833     /usr/ccs/lib/libform.so \
834     /usr/ccs/lib/llib-lform \
835     /usr/ccs/lib/llib-lform.ln \
836     /usr/ccs/lib/libgen.so \
837     /usr/ccs/lib/llib-lgen \
838     /usr/ccs/lib/llib-lgen.ln \
839     /usr/ccs/lib/libmalloc.so \
840     /usr/ccs/lib/libmenu.so \
841     /usr/ccs/lib/llib-lmenu \
842     /usr/ccs/lib/llib-lmenu.ln \
843     /usr/ccs/lib/libpanel.so \
844     /usr/ccs/lib/llib-lpanel \
845     /usr/ccs/lib/llib-lpanel.ln \
846     /usr/ccs/lib/libtermli.so \
847     /usr/ccs/lib/llib-ltermli \
848     /usr/ccs/lib/llib-ltermli.ln \
849     /usr/ccs/lib/libtermcap.so \

```

```

850     /usr/ccs/lib/llib-ltermcap \
851     /usr/ccs/lib/llib-ltermcap.ln \
852     /usr/ccs/lib/values-Xa.o \
853     /usr/ccs/lib/values-Xc.o \
854     /usr/ccs/lib/values-Xs.o \
855     /usr/ccs/lib/values-Xt.o \
856     /usr/ccs/lib/values-xpg4.o \
857     /usr/ccs/lib/values-xpg6.o \
858     /usr/ccs/lib/libl.so \
859     /usr/ccs/lib/llib-ll.ln \
860     /usr/ccs/lib/liby.so \
861     /usr/ccs/lib/llib-ly.ln \
862     /usr/lib/libp/libc.so.1 \
863     /usr/lib/lwp/libthread.so.1 \
864     /usr/lib/lwp/libthread_db.so.1

866 SYM.USRCCSLIB64= \
867     /usr/ccs/lib/$(MACH64)/libcurses.so \
868     /usr/ccs/lib/$(MACH64)/llib-lcurses.ln \
869     /usr/ccs/lib/$(MACH64)/libform.so \
870     /usr/ccs/lib/$(MACH64)/llib-lform.ln \
871     /usr/ccs/lib/$(MACH64)/libgen.so \
872     /usr/ccs/lib/$(MACH64)/llib-lgen.ln \
873     /usr/ccs/lib/$(MACH64)/libmalloc.so \
874     /usr/ccs/lib/$(MACH64)/libmenu.so \
875     /usr/ccs/lib/$(MACH64)/llib-lmenu.ln \
876     /usr/ccs/lib/$(MACH64)/libpanel.so \
877     /usr/ccs/lib/$(MACH64)/llib-lpanel.ln \
878     /usr/ccs/lib/$(MACH64)/libtermli.so \
879     /usr/ccs/lib/$(MACH64)/llib-ltermli.ln \
880     /usr/ccs/lib/$(MACH64)/libtermcap.so \
881     /usr/ccs/lib/$(MACH64)/llib-ltermcap.ln \
882     /usr/ccs/lib/$(MACH64)/values-Xa.o \
883     /usr/ccs/lib/$(MACH64)/values-Xc.o \
884     /usr/ccs/lib/$(MACH64)/values-Xs.o \
885     /usr/ccs/lib/$(MACH64)/values-Xt.o \
886     /usr/ccs/lib/$(MACH64)/values-xpg4.o \
887     /usr/ccs/lib/$(MACH64)/values-xpg6.o \
888     /usr/ccs/lib/$(MACH64)/libl.so \
889     /usr/ccs/lib/$(MACH64)/llib-ll.ln \
890     /usr/ccs/lib/$(MACH64)/liby.so \
891     /usr/ccs/lib/$(MACH64)/llib-ly.ln \
892     /usr/lib/libp/$(MACH64)/libc.so.1 \
893     /usr/lib/lwp/$(MACH64)/libthread.so.1 \
894     /usr/lib/lwp/$(MACH64)/libthread_db.so.1

896 # Special symlinks to direct libraries that have been moved
897 # from /usr/lib to /lib in order to live in the root filesystem.
898 $(ROOT)/lib/libposix4.so.1:= REALPATH=librt.so.1
899 $(ROOT)/lib/libposix4.so:= REALPATH=libposix4.so.1
900 $(ROOT)/lib/llib-lposix4:= REALPATH=llib-lrt
901 $(ROOT)/lib/liblwp/liblwp-libl.so.1:= REALPATH=liblwp-libl.so.1
902 $(ROOT)/lib/liblwp/liblwp-libc.so.1:= REALPATH=libc_db.so.1
903 $(ROOT)/lib/liblwp/liblwp-libc.so:= REALPATH=libc_db.so.1
904 $(ROOT)/usr/lib/libld.so.1:= REALPATH=../../../../lib/ld.so.1
905 $(ROOT)/usr/lib/libadm.so.1:= REALPATH=../../../../lib/libadm.so.1
906 $(ROOT)/usr/lib/libadm.so:= REALPATH=../../../../lib/libadm.so.1
907 $(ROOT)/usr/lib/libaio.so.1:= REALPATH=../../../../lib/libaio.so.1
908 $(ROOT)/usr/lib/libaio.so:= REALPATH=../../../../lib/libaio.so.1
909 $(ROOT)/usr/lib/libavl.so.1:= REALPATH=../../../../lib/libavl.so.1
910 $(ROOT)/usr/lib/libavl.so:= REALPATH=../../../../lib/libavl.so.1
911 $(ROOT)/usr/lib/libbsm.so.1:= REALPATH=../../../../lib/libbsm.so.1
912 $(ROOT)/usr/lib/libbsm.so:= REALPATH=../../../../lib/libbsm.so.1
913 $(ROOT)/usr/lib/libc.so.1:= REALPATH=../../../../lib/libc.so.1
914 $(ROOT)/usr/lib/libc.so:= REALPATH=../../../../lib/libc.so.1
915 $(ROOT)/usr/lib/libc_db.so.1:= REALPATH=../../../../lib/libc_db.so.1

```

```

916 $(ROOT)/usr/lib/libc_db.so:= REALPATH=.../lib/libc_db.so.1
917 $(ROOT)/usr/lib/libcmdutils.so.1:= REALPATH=.../lib/libcmdutils.so.1
918 $(ROOT)/usr/lib/libcmdutils.so:= REALPATH=.../lib/libcmdutils.so.1
919 $(ROOT)/usr/lib/libcontract.so.1:= REALPATH=.../lib/libcontract.so.1
920 $(ROOT)/usr/lib/libcontract.so:= REALPATH=.../lib/libcontract.so.1
921 $(ROOT)/usr/lib/libcryptoutil.so.1:= REALPATH=.../lib/libcryptoutil.so.1
922 $(ROOT)/usr/lib/libcryptoutil.so:= REALPATH=.../lib/libcryptoutil.so.1
923 $(ROOT)/usr/lib/libctf.so.1:= REALPATH=.../lib/libctf.so.1
924 $(ROOT)/usr/lib/libctf.so:= REALPATH=.../lib/libctf.so.1
925 $(ROOT)/usr/lib/libcurses.so.1:= REALPATH=.../lib/libcurses.so.1
926 $(ROOT)/usr/lib/libcurses.so:= REALPATH=.../lib/libcurses.so.1
927 $(ROOT)/usr/lib/libdevice.so.1:= REALPATH=.../lib/libdevice.so.1
928 $(ROOT)/usr/lib/libdevice.so:= REALPATH=.../lib/libdevice.so.1
929 $(ROOT)/usr/lib/libdevvid.so.1:= REALPATH=.../lib/libdevvid.so.1
930 $(ROOT)/usr/lib/libdevvid.so:= REALPATH=.../lib/libdevvid.so.1
931 $(ROOT)/usr/lib/libdevinfo.so.1:= REALPATH=.../lib/libdevinfo.so.1
932 $(ROOT)/usr/lib/libdevinfo.so:= REALPATH=.../lib/libdevinfo.so.1
933 $(ROOT)/usr/lib/libdhcpagent.so.1:= REALPATH=.../lib/libdhcpagent.so.1
934 $(ROOT)/usr/lib/libdhcpagent.so:= REALPATH=.../lib/libdhcpagent.so.1
935 $(ROOT)/usr/lib/libdhcputil.so.1:= REALPATH=.../lib/libdhcputil.so.1
936 $(ROOT)/usr/lib/libdhcputil.so:= REALPATH=.../lib/libdhcputil.so.1
937 $(ROOT)/usr/lib/libddl.so.1:= REALPATH=.../lib/libddl.so.1
938 $(ROOT)/usr/lib/libddl.so:= REALPATH=.../lib/libddl.so.1
939 $(ROOT)/usr/lib/libdlpi.so.1:= REALPATH=.../lib/libdlpi.so.1
940 $(ROOT)/usr/lib/libdlpi.so:= REALPATH=.../lib/libdlpi.so.1
941 $(ROOT)/usr/lib/libdoor.so.1:= REALPATH=.../lib/libdoor.so.1
942 $(ROOT)/usr/lib/libdoor.so:= REALPATH=.../lib/libdoor.so.1
943 $(ROOT)/usr/lib/libefi.so.1:= REALPATH=.../lib/libefi.so.1
944 $(ROOT)/usr/lib/libefi.so:= REALPATH=.../lib/libefi.so.1
945 $(ROOT)/usr/lib/libelf.so.1:= REALPATH=.../lib/libelf.so.1
946 $(ROOT)/usr/lib/libelf.so:= REALPATH=.../lib/libelf.so.1
947 $(ROOT)/usr/lib/libfdisk.so.1:= REALPATH=.../lib/libfdisk.so.1
948 $(ROOT)/usr/lib/libfdisk.so:= REALPATH=.../lib/libfdisk.so.1
949 $(ROOT)/usr/lib/libgen.so.1:= REALPATH=.../lib/libgen.so.1
950 $(ROOT)/usr/lib/libgen.so:= REALPATH=.../lib/libgen.so.1
951 $(ROOT)/usr/lib/libinetutil.so.1:= REALPATH=.../lib/libinetutil.so.1
952 $(ROOT)/usr/lib/libinetutil.so:= REALPATH=.../lib/libinetutil.so.1
953 $(ROOT)/usr/lib/libintl.so.1:= REALPATH=.../lib/libintl.so.1
954 $(ROOT)/usr/lib/libintl.so:= REALPATH=.../lib/libintl.so.1
955 $(ROOT)/usr/lib/libkmf.so.1:= REALPATH=.../lib/libkmf.so.1
956 $(ROOT)/usr/lib/libkmf.so:= REALPATH=.../lib/libkmf.so.1
957 $(ROOT)/usr/lib/libkmfberder.so.1:= REALPATH=.../lib/libkmfberder.so.1
958 $(ROOT)/usr/lib/libkmfberder.so:= REALPATH=.../lib/libkmfberder.so.1
959 $(ROOT)/usr/lib/libkstat.so.1:= REALPATH=.../lib/libkstat.so.1
960 $(ROOT)/usr/lib/libkstat.so:= REALPATH=.../lib/libkstat.so.1
961 $(ROOT)/usr/lib/liblddb.so.4:= REALPATH=.../lib/liblddb.so.4
962 $(ROOT)/usr/lib/libmd.so.1:= REALPATH=.../lib/libmd.so.1
963 $(ROOT)/usr/lib/libmd.so:= REALPATH=.../lib/libmd.so.1
964 $(ROOT)/usr/lib/libmd5.so.1:= REALPATH=.../lib/libmd5.so.1
965 $(ROOT)/usr/lib/libmd5.so:= REALPATH=.../lib/libmd5.so.1
966 $(ROOT)/usr/lib/libmeta.so.1:= REALPATH=.../lib/libmeta.so.1
967 $(ROOT)/usr/lib/libmeta.so:= REALPATH=.../lib/libmeta.so.1
968 $(ROOT)/usr/lib/libmp.so.1:= REALPATH=.../lib/libmp.so.1
969 $(ROOT)/usr/lib/libmp.so.2:= REALPATH=.../lib/libmp.so.2
970 $(ROOT)/usr/lib/libmp.so:= REALPATH=.../lib/libmp.so.2
971 $(ROOT)/usr/lib/libnsl.so.1:= REALPATH=.../lib/libnsl.so.1
972 $(ROOT)/usr/lib/libnsl.so:= REALPATH=.../lib/libnsl.so.1
973 $(ROOT)/usr/lib/libnvpair.so.1:= REALPATH=.../lib/libnvpair.so.1
974 $(ROOT)/usr/lib/libnvpair.so:= REALPATH=.../lib/libnvpair.so.1
975 $(ROOT)/usr/lib/libnvpam.so.1:= REALPATH=.../lib/libnvpam.so.1
976 $(ROOT)/usr/lib/libnvpam.so:= REALPATH=.../lib/libnvpam.so.1
977 $(ROOT)/usr/lib/libposix4.so.1:= REALPATH=.../lib/librt.so.1
978 $(ROOT)/usr/lib/libposix4.so:= REALPATH=.../lib/librt.so.1
979 $(ROOT)/usr/lib/libproc.so.1:= REALPATH=.../lib/libproc.so.1
980 $(ROOT)/usr/lib/libproc.so:= REALPATH=.../lib/libproc.so.1
981 $(ROOT)/usr/lib/libpthread.so.1:= REALPATH=.../lib/libpthread.so.1

```

```

982 $(ROOT)/usr/lib/libpthread.so:= REALPATH=.../lib/libpthread.so.1
983 $(ROOT)/usr/lib/librcm.so.1:= REALPATH=.../lib/librcm.so.1
984 $(ROOT)/usr/lib/librcm.so:= REALPATH=.../lib/librcm.so.1
985 $(ROOT)/usr/lib/libresolv.so.1:= REALPATH=.../lib/libresolv.so.1
986 $(ROOT)/usr/lib/libresolv.so.2:= REALPATH=.../lib/libresolv.so.2
987 $(ROOT)/usr/lib/libresolv.so:= REALPATH=.../lib/libresolv.so.2
988 $(ROOT)/usr/lib/librestart.so.1:= REALPATH=.../lib/librestart.so.1
989 $(ROOT)/usr/lib/librestart.so:= REALPATH=.../lib/librestart.so.1
990 $(ROOT)/usr/lib/librpcsvc.so.1:= REALPATH=.../lib/librpcsvc.so.1
991 $(ROOT)/usr/lib/librpcsvc.so:= REALPATH=.../lib/librpcsvc.so.1
992 $(ROOT)/usr/lib/librt.so.1:= REALPATH=.../lib/librt.so.1
993 $(ROOT)/usr/lib/librt.so:= REALPATH=.../lib/librt.so.1
994 $(ROOT)/usr/lib/librtld.so.1:= REALPATH=.../lib/librtld.so.1
995 $(ROOT)/usr/lib/librtld_db.so.1:= REALPATH=.../lib/librtld_db.so.1
996 $(ROOT)/usr/lib/librtld_db.so:= REALPATH=.../lib/librtld_db.so.1
997 $(ROOT)/usr/lib/libscf.so.1:= REALPATH=.../lib/libscf.so.1
998 $(ROOT)/usr/lib/libscf.so:= REALPATH=.../lib/libscf.so.1
999 $(ROOT)/usr/lib/libsec.so.1:= REALPATH=.../lib/libsec.so.1
1000 $(ROOT)/usr/lib/libsec.so:= REALPATH=.../lib/libsec.so.1
1001 $(ROOT)/usr/lib/libsecdb.so.1:= REALPATH=.../lib/libsecdb.so.1
1002 $(ROOT)/usr/lib/libsecdb.so:= REALPATH=.../lib/libsecdb.so.1
1003 $(ROOT)/usr/lib/libsendfile.so.1:= REALPATH=.../lib/libsendfile.so.1
1004 $(ROOT)/usr/lib/libsendfile.so:= REALPATH=.../lib/libsendfile.so.1
1005 $(ROOT)/usr/lib/libsocket.so.1:= REALPATH=.../lib/libsocket.so.1
1006 $(ROOT)/usr/lib/libsocket.so:= REALPATH=.../lib/libsocket.so.1
1007 $(ROOT)/usr/lib/libsysevent.so.1:= REALPATH=.../lib/libsyssevent.so.1
1008 $(ROOT)/usr/lib/libsyssevent.so:= REALPATH=.../lib/libsyssevent.so.1
1009 $(ROOT)/usr/lib/libtermcap.so.1:= REALPATH=.../lib/libtermcap.so.1
1010 $(ROOT)/usr/lib/libtermcap.so:= REALPATH=.../lib/libtermcap.so.1
1011 $(ROOT)/usr/lib/libtermlib.so.1:= REALPATH=.../lib/libcurses.so.1
1012 $(ROOT)/usr/lib/libtermlib.so:= REALPATH=.../lib/libcurses.so.1
1013 $(ROOT)/usr/lib/libthread.so.1:= REALPATH=.../lib/libthread.so.1
1014 $(ROOT)/usr/lib/libthread.so:= REALPATH=.../lib/libthread.so.1
1015 $(ROOT)/usr/lib/libthread_db.so.1:= REALPATH=.../lib/libc_db.so.1
1016 $(ROOT)/usr/lib/libthread_db.so:= REALPATH=.../lib/libc_db.so.1
1017 $(ROOT)/usr/lib/libtsnet.so.1:= REALPATH=.../lib/libtsnet.so.1
1018 $(ROOT)/usr/lib/libtsnet.so:= REALPATH=.../lib/libtsnet.so.1
1019 $(ROOT)/usr/lib/libtsol.so.2:= REALPATH=.../lib/libtsol.so.2
1020 $(ROOT)/usr/lib/libtsol.so:= REALPATH=.../lib/libtsol.so.2
1021 $(ROOT)/usr/lib/libumem.so.1:= REALPATH=.../lib/libumem.so.1
1022 $(ROOT)/usr/lib/libumem.so:= REALPATH=.../lib/libumem.so.1
1023 $(ROOT)/usr/lib/libuuid.so.1:= REALPATH=.../lib/libuuid.so.1
1024 $(ROOT)/usr/lib/libuuid.so:= REALPATH=.../lib/libuuid.so.1
1025 $(ROOT)/usr/lib/libuutil.so.1:= REALPATH=.../lib/libuutil.so.1
1026 $(ROOT)/usr/lib/libuutil.so:= REALPATH=.../lib/libuutil.so.1
1027 $(ROOT)/usr/lib/libw.so.1:= REALPATH=.../lib/libw.so.1
1028 $(ROOT)/usr/lib/libw.so:= REALPATH=.../lib/libw.so.1
1029 $(ROOT)/usr/lib/libxnet.so.1:= REALPATH=.../lib/libxnet.so.1
1030 $(ROOT)/usr/lib/libxnet.so:= REALPATH=.../lib/libxnet.so.1
1031 $(ROOT)/usr/lib/libzfs.so.1:= REALPATH=.../lib/libzfs.so.1
1032 $(ROOT)/usr/lib/libzfs.so:= REALPATH=.../lib/libzfs.so.1
1033 $(ROOT)/usr/lib/libzfs_core.so.1:= REALPATH=.../lib/libzfs_core.so.1
1034 $(ROOT)/usr/lib/libzfs_core.so:= REALPATH=.../lib/libzfs_core.so.1
1035 $(ROOT)/usr/lib/lib-ladm.ln:= REALPATH=.../lib/lib-ladm.ln
1036 $(ROOT)/usr/lib/lib-ladm:= REALPATH=.../lib/lib-ladm
1037 $(ROOT)/usr/lib/lib-laic.ln:= REALPATH=.../lib/lib-laic.ln
1038 $(ROOT)/usr/lib/lib-laic:= REALPATH=.../lib/lib-laic
1039 $(ROOT)/usr/lib/lib-lavl.ln:= REALPATH=.../lib/lib-lavl.ln
1040 $(ROOT)/usr/lib/lib-lavl:= REALPATH=.../lib/lib-lavl
1041 $(ROOT)/usr/lib/lib-lbms.ln:= REALPATH=.../lib/lib-lbms.ln
1042 $(ROOT)/usr/lib/lib-lbms:= REALPATH=.../lib/lib-lbms
1043 $(ROOT)/usr/lib/lib-lc.ln:= REALPATH=.../lib/lib-lc.ln
1044 $(ROOT)/usr/lib/lib-lc:= REALPATH=.../lib/lib-lc
1045 $(ROOT)/usr/lib/lib-lcmdutils.ln:= REALPATH=.../lib/lib-lcmdutils.ln
1046 $(ROOT)/usr/lib/lib-lcmdutils:= REALPATH=.../lib/lib-lcmdutils
1047 $(ROOT)/usr/lib/lib-lcontract.ln:= REALPATH=.../lib/lib-lcontract.ln

```



```

1048 $(ROOT)/usr/lib/lib-lcontract:= REALPATH=../lib/lib-lcontract
1049 $(ROOT)/usr/lib/lib-lctf.ln:= REALPATH=../lib/lib-lctf.ln
1050 $(ROOT)/usr/lib/lib-lctf:= REALPATH=../lib/lib-lctf
1051 $(ROOT)/usr/lib/lib-lcurses.ln:= REALPATH=../lib/lib-lcurses.ln
1052 $(ROOT)/usr/lib/lib-lcurses:= REALPATH=../lib/lib-lcurses
1053 $(ROOT)/usr/lib/lib-ldevice.ln:= REALPATH=../lib/lib-ldevice.ln
1054 $(ROOT)/usr/lib/lib-ldevice:= REALPATH=../lib/lib-ldevice
1055 $(ROOT)/usr/lib/lib-ldevvid.ln:= REALPATH=../lib/lib-ldevvid.ln
1056 $(ROOT)/usr/lib/lib-ldevvid:= REALPATH=../lib/lib-ldevvid
1057 $(ROOT)/usr/lib/lib-ldevinfo.ln:= REALPATH=../lib/lib-ldevinfo.ln
1058 $(ROOT)/usr/lib/lib-ldevinfo:= REALPATH=../lib/lib-ldevinfo
1059 $(ROOT)/usr/lib/lib-ldhcpagent.ln:= REALPATH=../lib/lib-ldhcpagent.ln
1060 $(ROOT)/usr/lib/lib-ldhcpagent:= REALPATH=../lib/lib-ldhcpagent
1061 $(ROOT)/usr/lib/lib-ldhcputil.ln:= REALPATH=../lib/lib-ldhcputil.ln
1062 $(ROOT)/usr/lib/lib-ldhcputil:= REALPATH=../lib/lib-ldhcputil
1063 $(ROOT)/usr/lib/lib-ldl.ln:= REALPATH=../lib/lib-ldl.ln
1064 $(ROOT)/usr/lib/lib-ldl:= REALPATH=../lib/lib-ldl
1065 $(ROOT)/usr/lib/lib-ldoor.ln:= REALPATH=../lib/lib-ldoor.ln
1066 $(ROOT)/usr/lib/lib-ldoor:= REALPATH=../lib/lib-ldoor
1067 $(ROOT)/usr/lib/lib-lefi.ln:= REALPATH=../lib/lib-lefi.ln
1068 $(ROOT)/usr/lib/lib-lefi:= REALPATH=../lib/lib-lefi
1069 $(ROOT)/usr/lib/lib-lelf.ln:= REALPATH=../lib/lib-lelf.ln
1070 $(ROOT)/usr/lib/lib-lelf:= REALPATH=../lib/lib-lelf
1071 $(ROOT)/usr/lib/lib-lfdisk.ln:= REALPATH=../lib/lib-lfdisk.ln
1072 $(ROOT)/usr/lib/lib-lfdisk:= REALPATH=../lib/lib-lfdisk
1073 $(ROOT)/usr/lib/lib-lgen.ln:= REALPATH=../lib/lib-lgen.ln
1074 $(ROOT)/usr/lib/lib-lgen:= REALPATH=../lib/lib-lgen
1075 $(ROOT)/usr/lib/lib-linetutil.ln:= REALPATH=../lib/lib-linetutil.ln
1076 $(ROOT)/usr/lib/lib-linetutil:= REALPATH=../lib/lib-linetutil
1077 $(ROOT)/usr/lib/lib-lintl.ln:= REALPATH=../lib/lib-lintl.ln
1078 $(ROOT)/usr/lib/lib-lintl:= REALPATH=../lib/lib-lintl
1079 $(ROOT)/usr/lib/lib-lkstat.ln:= REALPATH=../lib/lib-lkstat.ln
1080 $(ROOT)/usr/lib/lib-lkstat:= REALPATH=../lib/lib-lkstat
1081 $(ROOT)/usr/lib/lib-lmd5.ln:= REALPATH=../lib/lib-lmd5.ln
1082 $(ROOT)/usr/lib/lib-lmd5:= REALPATH=../lib/lib-lmd5
1083 $(ROOT)/usr/lib/lib-lmeta.ln:= REALPATH=../lib/lib-lmeta.ln
1084 $(ROOT)/usr/lib/lib-lmeta:= REALPATH=../lib/lib-lmeta
1085 $(ROOT)/usr/lib/lib-lns1.ln:= REALPATH=../lib/lib-lns1.ln
1086 $(ROOT)/usr/lib/lib-lns1:= REALPATH=../lib/lib-lns1
1087 $(ROOT)/usr/lib/lib-lnvpair.ln:= REALPATH=../lib/lib-lnvpair.ln
1088 $(ROOT)/usr/lib/lib-lnvpair:= REALPATH=../lib/lib-lnvpair
1089 $(ROOT)/usr/lib/lib-lpam.ln:= REALPATH=../lib/lib-lpam.ln
1090 $(ROOT)/usr/lib/lib-lpam:= REALPATH=../lib/lib-lpam
1091 $(ROOT)/usr/lib/lib-lposix4.ln:= REALPATH=../lib/lib-lrt.ln
1092 $(ROOT)/usr/lib/lib-lposix4:= REALPATH=../lib/lib-lrt
1093 $(ROOT)/usr/lib/lib-lpthread.ln:= REALPATH=../lib/lib-lpthread.ln
1094 $(ROOT)/usr/lib/lib-lpthread:= REALPATH=../lib/lib-lpthread
1095 $(ROOT)/usr/lib/lib-lresolv.ln:= REALPATH=../lib/lib-lresolv.ln
1096 $(ROOT)/usr/lib/lib-lresolv:= REALPATH=../lib/lib-lresolv
1097 $(ROOT)/usr/lib/lib-lrpcsvc.ln:= REALPATH=../lib/lib-lrpcsvc.ln
1098 $(ROOT)/usr/lib/lib-lrpcsvc:= REALPATH=../lib/lib-lrpcsvc
1099 $(ROOT)/usr/lib/lib-lrt.ln:= REALPATH=../lib/lib-lrt.ln
1100 $(ROOT)/usr/lib/lib-lrt:= REALPATH=../lib/lib-lrt
1101 $(ROOT)/usr/lib/lib-lrtld_db.ln:= REALPATH=../lib/lib-lrtld_db.ln
1102 $(ROOT)/usr/lib/lib-lrtld_db:= REALPATH=../lib/lib-lrtld_db
1103 $(ROOT)/usr/lib/lib-lscf.ln:= REALPATH=../lib/lib-lscf.ln
1104 $(ROOT)/usr/lib/lib-lscf:= REALPATH=../lib/lib-lscf
1105 $(ROOT)/usr/lib/lib-lsec.ln:= REALPATH=../lib/lib-lsec.ln
1106 $(ROOT)/usr/lib/lib-lsec:= REALPATH=../lib/lib-lsec
1107 $(ROOT)/usr/lib/lib-lsecdb.ln:= REALPATH=../lib/lib-lsecdb.ln
1108 $(ROOT)/usr/lib/lib-lsecdb:= REALPATH=../lib/lib-lsecdb
1109 $(ROOT)/usr/lib/lib-lsendfile.ln:= REALPATH=../lib/lib-lsendfile.ln
1110 $(ROOT)/usr/lib/lib-lsendfile:= REALPATH=../lib/lib-lsendfile
1111 $(ROOT)/usr/lib/lib-lsocket.ln:= REALPATH=../lib/lib-lsocket.ln
1112 $(ROOT)/usr/lib/lib-lsocket:= REALPATH=../lib/lib-lsocket
1113 $(ROOT)/usr/lib/lib-lsysevent.ln:= REALPATH=../lib/lib-lsysevent.ln

```

```

1114 $(ROOT)/usr/lib/lib-lsysevent:= REALPATH=../lib/lib-lsysevent
1115 $(ROOT)/usr/lib/lib-ltermcap.ln:= REALPATH=../lib/lib-ltermcap.ln
1116 $(ROOT)/usr/lib/lib-ltermcap:= REALPATH=../lib/lib-ltermcap
1117 $(ROOT)/usr/lib/lib-ltermlib.ln:= REALPATH=../lib/lib-ltermlib.ln
1118 $(ROOT)/usr/lib/lib-ltermlib:= REALPATH=../lib/lib-ltermlib
1119 $(ROOT)/usr/lib/lib-lthread.ln:= REALPATH=../lib/lib-lthread.ln
1120 $(ROOT)/usr/lib/lib-lthread:= REALPATH=../lib/lib-lthread
1121 $(ROOT)/usr/lib/lib-lthread_db.ln:= REALPATH=../lib/lib-lc_db.ln
1122 $(ROOT)/usr/lib/lib-lthread_db:= REALPATH=../lib/lib-lc_db
1123 $(ROOT)/usr/lib/lib-ltsnet.ln:= REALPATH=../lib/lib-ltsnet.ln
1124 $(ROOT)/usr/lib/lib-ltsnet:= REALPATH=../lib/lib-ltsnet
1125 $(ROOT)/usr/lib/lib-ltsol.ln:= REALPATH=../lib/lib-ltsol.ln
1126 $(ROOT)/usr/lib/lib-ltsol:= REALPATH=../lib/lib-ltsol
1127 $(ROOT)/usr/lib/lib-lumem.ln:= REALPATH=../lib/lib-lumem.ln
1128 $(ROOT)/usr/lib/lib-lumem:= REALPATH=../lib/lib-lumem
1129 $(ROOT)/usr/lib/lib-luuid.ln:= REALPATH=../lib/lib-luuid.ln
1130 $(ROOT)/usr/lib/lib-luuid:= REALPATH=../lib/lib-luuid
1131 $(ROOT)/usr/lib/lib-lxnet.ln:= REALPATH=../lib/lib-lxnet.ln
1132 $(ROOT)/usr/lib/lib-lxnet:= REALPATH=../lib/lib-lxnet
1133 $(ROOT)/usr/lib/lib-lzfs.ln:= REALPATH=../lib/lib-lzfs.ln
1134 $(ROOT)/usr/lib/lib-lzfs:= REALPATH=../lib/lib-lzfs
1135 $(ROOT)/usr/lib/lib-lzfs_core.ln:= REALPATH=../lib/lib-lzfs_core.ln
1136 $(ROOT)/usr/lib/lib-lzfs_core:= REALPATH=../lib/lib-lzfs_core
1137 $(ROOT)/usr/lib/nss_compat.so.1:= REALPATH=../lib/nss_compat.so.1
1138 $(ROOT)/usr/lib/nss_dns.so.1:= REALPATH=../lib/nss_dns.so.1
1139 $(ROOT)/usr/lib/nss_files.so.1:= REALPATH=../lib/nss_files.so.1
1140 $(ROOT)/usr/lib/nss_nis.so.1:= REALPATH=../lib/nss_nis.so.1
1141 $(ROOT)/usr/lib/nss_user.so.1:= REALPATH=../lib/nss_user.so.1
1142 $(ROOT)/usr/lib/fm/libfmevent.so.1:= REALPATH=../lib/fm/libfmevent.so.1
1143 $(ROOT)/usr/lib/fm/libfmevent.so:= REALPATH=../lib/fm/libfmevent.so.1
1144 $(ROOT)/usr/lib/fm/lib-lfmevent.ln:= REALPATH=../lib/fm/lib-lfmevent.ln
1145 $(ROOT)/usr/lib/fm/lib-lfmevent:= REALPATH=../lib/fm/lib-lfmevent

1147 $(ROOT)/lib/$(MACH64)/libposix4.so.1:= \
1148 REALPATH=librt.so.1
1149 $(ROOT)/lib/$(MACH64)/libposix4.so:= \
1150 REALPATH=libposix4.so.1
1151 $(ROOT)/lib/$(MACH64)/lib-lposix4.ln:= \
1152 REALPATH=lib-lrt.ln
1153 $(ROOT)/lib/$(MACH64)/libthread_db.so.1:= \
1154 REALPATH=libc_db.so.1
1155 $(ROOT)/lib/$(MACH64)/libthread_db.so:= \
1156 REALPATH=libc_db.so.1
1157 $(ROOT)/usr/lib/$(MACH64)/ld.so.1:= \
1158 REALPATH=../lib/$(MACH64)/ld.so.1
1159 $(ROOT)/usr/lib/$(MACH64)/libadm.so.1:= \
1160 REALPATH=../lib/$(MACH64)/libadm.so.1
1161 $(ROOT)/usr/lib/$(MACH64)/libadm.so:= \
1162 REALPATH=../lib/$(MACH64)/libadm.so.1
1163 $(ROOT)/usr/lib/$(MACH64)/libaio.so.1:= \
1164 REALPATH=../lib/$(MACH64)/libaio.so.1
1165 $(ROOT)/usr/lib/$(MACH64)/libaio.so:= \
1166 REALPATH=../lib/$(MACH64)/libaio.so.1
1167 $(ROOT)/usr/lib/$(MACH64)/libavl.so.1:= \
1168 REALPATH=../lib/$(MACH64)/libavl.so.1
1169 $(ROOT)/usr/lib/$(MACH64)/libavl.so:= \
1170 REALPATH=../lib/$(MACH64)/libavl.so.1
1171 $(ROOT)/usr/lib/$(MACH64)/libbsm.so.1:= \
1172 REALPATH=../lib/$(MACH64)/libbsm.so.1
1173 $(ROOT)/usr/lib/$(MACH64)/libbsm.so:= \
1174 REALPATH=../lib/$(MACH64)/libbsm.so.1
1175 $(ROOT)/usr/lib/$(MACH64)/libc.so.1:= \
1176 REALPATH=../lib/$(MACH64)/libc.so.1
1177 $(ROOT)/usr/lib/$(MACH64)/libc.so:= \
1178 REALPATH=../lib/$(MACH64)/libc.so.1
1179 $(ROOT)/usr/lib/$(MACH64)/libc_db.so.1:= \

```

```

1180 REALPATH=../../../../lib/$(MACH64)/libc_db.so.1
1181 $(ROOT)/usr/lib/$(MACH64)/libc_db.so:= \
1182 REALPATH=../../../../lib/$(MACH64)/libc_db.so.1
1183 $(ROOT)/usr/lib/$(MACH64)/libcmdutils.so.1:= \
1184 REALPATH=../../../../lib/$(MACH64)/libcmdutils.so.1
1185 $(ROOT)/usr/lib/$(MACH64)/libcmdutils.so:= \
1186 REALPATH=../../../../lib/$(MACH64)/libcmdutils.so.1
1187 $(ROOT)/usr/lib/$(MACH64)/libcontract.so.1:= \
1188 REALPATH=../../../../lib/$(MACH64)/libcontract.so.1
1189 $(ROOT)/usr/lib/$(MACH64)/libcontract.so:= \
1190 REALPATH=../../../../lib/$(MACH64)/libcontract.so.1
1191 $(ROOT)/usr/lib/$(MACH64)/libctf.so.1:= \
1192 REALPATH=../../../../lib/$(MACH64)/libctf.so.1
1193 $(ROOT)/usr/lib/$(MACH64)/libctf.so:= \
1194 REALPATH=../../../../lib/$(MACH64)/libctf.so.1
1195 $(ROOT)/usr/lib/$(MACH64)/libcurses.so.1:= \
1196 REALPATH=../../../../lib/$(MACH64)/libcurses.so.1
1197 $(ROOT)/usr/lib/$(MACH64)/libcurses.so:= \
1198 REALPATH=../../../../lib/$(MACH64)/libcurses.so.1
1199 $(ROOT)/usr/lib/$(MACH64)/libdevice.so.1:= \
1200 REALPATH=../../../../lib/$(MACH64)/libdevice.so.1
1201 $(ROOT)/usr/lib/$(MACH64)/libdevice.so:= \
1202 REALPATH=../../../../lib/$(MACH64)/libdevice.so.1
1203 $(ROOT)/usr/lib/$(MACH64)/libdevvid.so.1:= \
1204 REALPATH=../../../../lib/$(MACH64)/libdevvid.so.1
1205 $(ROOT)/usr/lib/$(MACH64)/libdevvid.so:= \
1206 REALPATH=../../../../lib/$(MACH64)/libdevvid.so.1
1207 $(ROOT)/usr/lib/$(MACH64)/libdevinfo.so.1:= \
1208 REALPATH=../../../../lib/$(MACH64)/libdevinfo.so.1
1209 $(ROOT)/usr/lib/$(MACH64)/libdevinfo.so:= \
1210 REALPATH=../../../../lib/$(MACH64)/libdevinfo.so.1
1211 $(ROOT)/usr/lib/$(MACH64)/libdhcputil.so.1:= \
1212 REALPATH=../../../../lib/$(MACH64)/libdhcputil.so.1
1213 $(ROOT)/usr/lib/$(MACH64)/libdhcputil.so:= \
1214 REALPATH=../../../../lib/$(MACH64)/libdhcputil.so.1
1215 $(ROOT)/usr/lib/$(MACH64)/libdl.so.1:= \
1216 REALPATH=../../../../lib/$(MACH64)/libdl.so.1
1217 $(ROOT)/usr/lib/$(MACH64)/libdl.so:= \
1218 REALPATH=../../../../lib/$(MACH64)/libdl.so.1
1219 $(ROOT)/usr/lib/$(MACH64)/libdlpi.so.1:= \
1220 REALPATH=../../../../lib/$(MACH64)/libdlpi.so.1
1221 $(ROOT)/usr/lib/$(MACH64)/libdlpi.so:= \
1222 REALPATH=../../../../lib/$(MACH64)/libdlpi.so.1
1223 $(ROOT)/usr/lib/$(MACH64)/libdoor.so.1:= \
1224 REALPATH=../../../../lib/$(MACH64)/libdoor.so.1
1225 $(ROOT)/usr/lib/$(MACH64)/libdoor.so:= \
1226 REALPATH=../../../../lib/$(MACH64)/libdoor.so.1
1227 $(ROOT)/usr/lib/$(MACH64)/libefi.so.1:= \
1228 REALPATH=../../../../lib/$(MACH64)/libefi.so.1
1229 $(ROOT)/usr/lib/$(MACH64)/libefi.so:= \
1230 REALPATH=../../../../lib/$(MACH64)/libefi.so.1
1231 $(ROOT)/usr/lib/$(MACH64)/libelf.so.1:= \
1232 REALPATH=../../../../lib/$(MACH64)/libelf.so.1
1233 $(ROOT)/usr/lib/$(MACH64)/libelf.so:= \
1234 REALPATH=../../../../lib/$(MACH64)/libelf.so.1
1235 $(ROOT)/usr/lib/$(MACH64)/libgen.so.1:= \
1236 REALPATH=../../../../lib/$(MACH64)/libgen.so.1
1237 $(ROOT)/usr/lib/$(MACH64)/libgen.so:= \
1238 REALPATH=../../../../lib/$(MACH64)/libgen.so.1
1239 $(ROOT)/usr/lib/$(MACH64)/libinetutil.so.1:= \
1240 REALPATH=../../../../lib/$(MACH64)/libinetutil.so.1
1241 $(ROOT)/usr/lib/$(MACH64)/libinetutil.so:= \
1242 REALPATH=../../../../lib/$(MACH64)/libinetutil.so.1
1243 $(ROOT)/usr/lib/$(MACH64)/libintl.so.1:= \
1244 REALPATH=../../../../lib/$(MACH64)/libintl.so.1
1245 $(ROOT)/usr/lib/$(MACH64)/libintl.so:= \

```

```

1246 REALPATH=../../../../lib/$(MACH64)/libintl.so.1
1247 $(ROOT)/usr/lib/$(MACH64)/libkstat.so.1:= \
1248 REALPATH=../../../../lib/$(MACH64)/libkstat.so.1
1249 $(ROOT)/usr/lib/$(MACH64)/libkstat.so:= \
1250 REALPATH=../../../../lib/$(MACH64)/libkstat.so.1
1251 $(ROOT)/usr/lib/$(MACH64)/liblddbg.so.4:= \
1252 REALPATH=../../../../lib/$(MACH64)/liblddbg.so.4
1253 $(ROOT)/usr/lib/$(MACH64)/libmd.so.1:= \
1254 REALPATH=../../../../lib/$(MACH64)/libmd.so.1
1255 $(ROOT)/usr/lib/$(MACH64)/libmd.so:= \
1256 REALPATH=../../../../lib/$(MACH64)/libmd.so.1
1257 $(ROOT)/usr/lib/$(MACH64)/libmd5.so.1:= \
1258 REALPATH=../../../../lib/$(MACH64)/libmd5.so.1
1259 $(ROOT)/usr/lib/$(MACH64)/libmd5.so:= \
1260 REALPATH=../../../../lib/$(MACH64)/libmd5.so.1
1261 $(ROOT)/usr/lib/$(MACH64)/libmp.so.2:= \
1262 REALPATH=../../../../lib/$(MACH64)/libmp.so.2
1263 $(ROOT)/usr/lib/$(MACH64)/libmp.so:= \
1264 REALPATH=../../../../lib/$(MACH64)/libmp.so.2
1265 $(ROOT)/usr/lib/$(MACH64)/libnsl.so.1:= \
1266 REALPATH=../../../../lib/$(MACH64)/libnsl.so.1
1267 $(ROOT)/usr/lib/$(MACH64)/libnsl.so:= \
1268 REALPATH=../../../../lib/$(MACH64)/libnsl.so.1
1269 $(ROOT)/usr/lib/$(MACH64)/libnvpair.so.1:= \
1270 REALPATH=../../../../lib/$(MACH64)/libnvpair.so.1
1271 $(ROOT)/usr/lib/$(MACH64)/libnvpair.so:= \
1272 REALPATH=../../../../lib/$(MACH64)/libnvpair.so.1
1273 $(ROOT)/usr/lib/$(MACH64)/libpam.so.1:= \
1274 REALPATH=../../../../lib/$(MACH64)/libpam.so.1
1275 $(ROOT)/usr/lib/$(MACH64)/libpam.so:= \
1276 REALPATH=../../../../lib/$(MACH64)/libpam.so.1
1277 $(ROOT)/usr/lib/$(MACH64)/libposix4.so.1:= \
1278 REALPATH=../../../../lib/$(MACH64)/libposix4.so.1
1279 $(ROOT)/usr/lib/$(MACH64)/libposix4.so:= \
1280 REALPATH=../../../../lib/$(MACH64)/librt.so.1
1281 $(ROOT)/usr/lib/$(MACH64)/libproc.so.1:= \
1282 REALPATH=../../../../lib/$(MACH64)/libproc.so.1
1283 $(ROOT)/usr/lib/$(MACH64)/libproc.so:= \
1284 REALPATH=../../../../lib/$(MACH64)/libproc.so.1
1285 $(ROOT)/usr/lib/$(MACH64)/libpthread.so.1:= \
1286 REALPATH=../../../../lib/$(MACH64)/libpthread.so.1
1287 $(ROOT)/usr/lib/$(MACH64)/libpthread.so:= \
1288 REALPATH=../../../../lib/$(MACH64)/libpthread.so.1
1289 $(ROOT)/usr/lib/$(MACH64)/librcm.so.1:= \
1290 REALPATH=../../../../lib/$(MACH64)/librcm.so.1
1291 $(ROOT)/usr/lib/$(MACH64)/librcm.so:= \
1292 REALPATH=../../../../lib/$(MACH64)/librcm.so.1
1293 $(ROOT)/usr/lib/$(MACH64)/libresolv.so.2:= \
1294 REALPATH=../../../../lib/$(MACH64)/libresolv.so.2
1295 $(ROOT)/usr/lib/$(MACH64)/libresolv.so:= \
1296 REALPATH=../../../../lib/$(MACH64)/libresolv.so.2
1297 $(ROOT)/usr/lib/$(MACH64)/librestart.so.1:= \
1298 REALPATH=../../../../lib/$(MACH64)/librestart.so.1
1299 $(ROOT)/usr/lib/$(MACH64)/librestart.so:= \
1300 REALPATH=../../../../lib/$(MACH64)/librestart.so.1
1301 $(ROOT)/usr/lib/$(MACH64)/librpcsvc.so.1:= \
1302 REALPATH=../../../../lib/$(MACH64)/librpcsvc.so.1
1303 $(ROOT)/usr/lib/$(MACH64)/librpcsvc.so:= \
1304 REALPATH=../../../../lib/$(MACH64)/librpcsvc.so.1
1305 $(ROOT)/usr/lib/$(MACH64)/librt.so.1:= \
1306 REALPATH=../../../../lib/$(MACH64)/librt.so.1
1307 $(ROOT)/usr/lib/$(MACH64)/librt.so:= \
1308 REALPATH=../../../../lib/$(MACH64)/librt.so.1
1309 $(ROOT)/usr/lib/$(MACH64)/librtld.so.1:= \
1310 REALPATH=../../../../lib/$(MACH64)/librtld.so.1
1311 $(ROOT)/usr/lib/$(MACH64)/librtld_db.so.1:= \

```

```

1312 REALPATH=../../../../lib/$(MACH64)/librtld_db.so.1
1313 $(ROOT)/usr/lib/$(MACH64)/librtld_db.so:= \
1314 REALPATH=../../../../lib/$(MACH64)/librtld_db.so.1
1315 $(ROOT)/usr/lib/$(MACH64)/libscf.so.1:= \
1316 REALPATH=../../../../lib/$(MACH64)/libscf.so.1
1317 $(ROOT)/usr/lib/$(MACH64)/libscf.so:= \
1318 REALPATH=../../../../lib/$(MACH64)/libscf.so.1
1319 $(ROOT)/usr/lib/$(MACH64)/libsec.so.1:= \
1320 REALPATH=../../../../lib/$(MACH64)/libsec.so.1
1321 $(ROOT)/usr/lib/$(MACH64)/libsec.so:= \
1322 REALPATH=../../../../lib/$(MACH64)/libsec.so.1
1323 $(ROOT)/usr/lib/$(MACH64)/libsecdb.so.1:= \
1324 REALPATH=../../../../lib/$(MACH64)/libsecdb.so.1
1325 $(ROOT)/usr/lib/$(MACH64)/libsecdb.so:= \
1326 REALPATH=../../../../lib/$(MACH64)/libsecdb.so.1
1327 $(ROOT)/usr/lib/$(MACH64)/libsendfile.so.1:= \
1328 REALPATH=../../../../lib/$(MACH64)/libsendfile.so.1
1329 $(ROOT)/usr/lib/$(MACH64)/libsendfile.so:= \
1330 REALPATH=../../../../lib/$(MACH64)/libsendfile.so.1
1331 $(ROOT)/usr/lib/$(MACH64)/libsocket.so.1:= \
1332 REALPATH=../../../../lib/$(MACH64)/libsocket.so.1
1333 $(ROOT)/usr/lib/$(MACH64)/libsocket.so:= \
1334 REALPATH=../../../../lib/$(MACH64)/libsocket.so.1
1335 $(ROOT)/usr/lib/$(MACH64)/libsysevent.so.1:= \
1336 REALPATH=../../../../lib/$(MACH64)/libsysevent.so.1
1337 $(ROOT)/usr/lib/$(MACH64)/libsysevent.so:= \
1338 REALPATH=../../../../lib/$(MACH64)/libsysevent.so.1
1339 $(ROOT)/usr/lib/$(MACH64)/libtermcap.so.1:= \
1340 REALPATH=../../../../lib/$(MACH64)/libtermcap.so.1
1341 $(ROOT)/usr/lib/$(MACH64)/libtermcap.so:= \
1342 REALPATH=../../../../lib/$(MACH64)/libtermcap.so.1
1343 $(ROOT)/usr/lib/$(MACH64)/libtermplib.so.1:= \
1344 REALPATH=../../../../lib/$(MACH64)/libcurses.so.1
1345 $(ROOT)/usr/lib/$(MACH64)/libtermplib.so:= \
1346 REALPATH=../../../../lib/$(MACH64)/libcurses.so.1
1347 $(ROOT)/usr/lib/$(MACH64)/libthread.so.1:= \
1348 REALPATH=../../../../lib/$(MACH64)/libthread.so.1
1349 $(ROOT)/usr/lib/$(MACH64)/libthread.so:= \
1350 REALPATH=../../../../lib/$(MACH64)/libthread.so.1
1351 $(ROOT)/usr/lib/$(MACH64)/libthread_db.so.1:= \
1352 REALPATH=../../../../lib/$(MACH64)/libc_db.so.1
1353 $(ROOT)/usr/lib/$(MACH64)/libthread_db.so:= \
1354 REALPATH=../../../../lib/$(MACH64)/libc_db.so.1
1355 $(ROOT)/usr/lib/$(MACH64)/libtsnet.so.1:= \
1356 REALPATH=../../../../lib/$(MACH64)/libtsnet.so.1
1357 $(ROOT)/usr/lib/$(MACH64)/libtsnet.so:= \
1358 REALPATH=../../../../lib/$(MACH64)/libtsnet.so.1
1359 $(ROOT)/usr/lib/$(MACH64)/libtsol.so.2:= \
1360 REALPATH=../../../../lib/$(MACH64)/libtsol.so.2
1361 $(ROOT)/usr/lib/$(MACH64)/libtsol.so:= \
1362 REALPATH=../../../../lib/$(MACH64)/libtsol.so.2
1363 $(ROOT)/usr/lib/$(MACH64)/libumem.so.1:= \
1364 REALPATH=../../../../lib/$(MACH64)/libumem.so.1
1365 $(ROOT)/usr/lib/$(MACH64)/libumem.so:= \
1366 REALPATH=../../../../lib/$(MACH64)/libumem.so.1
1367 $(ROOT)/usr/lib/$(MACH64)/libuuid.so.1:= \
1368 REALPATH=../../../../lib/$(MACH64)/libuuid.so.1
1369 $(ROOT)/usr/lib/$(MACH64)/libuuid.so:= \
1370 REALPATH=../../../../lib/$(MACH64)/libuuid.so.1
1371 $(ROOT)/usr/lib/$(MACH64)/libuutil.so.1:= \
1372 REALPATH=../../../../lib/$(MACH64)/libuutil.so.1
1373 $(ROOT)/usr/lib/$(MACH64)/libuutil.so:= \
1374 REALPATH=../../../../lib/$(MACH64)/libuutil.so.1
1375 $(ROOT)/usr/lib/$(MACH64)/libw.so.1:= \
1376 REALPATH=../../../../lib/$(MACH64)/libw.so.1
1377 $(ROOT)/usr/lib/$(MACH64)/libw.so:= \

```

```

1378 REALPATH=../../../../lib/$(MACH64)/libw.so.1
1379 $(ROOT)/usr/lib/$(MACH64)/libxnet.so.1:= \
1380 REALPATH=../../../../lib/$(MACH64)/libxnet.so.1
1381 $(ROOT)/usr/lib/$(MACH64)/libxnet.so:= \
1382 REALPATH=../../../../lib/$(MACH64)/libxnet.so.1
1383 $(ROOT)/usr/lib/$(MACH64)/libzfs.so:= \
1384 REALPATH=../../../../lib/$(MACH64)/libzfs.so.1
1385 $(ROOT)/usr/lib/$(MACH64)/libzfs.so.1:= \
1386 REALPATH=../../../../lib/$(MACH64)/libzfs.so.1
1387 $(ROOT)/usr/lib/$(MACH64)/libzfs_core.so:= \
1388 REALPATH=../../../../lib/$(MACH64)/libzfs_core.so.1
1389 $(ROOT)/usr/lib/$(MACH64)/libzfs_core.so.1:= \
1390 REALPATH=../../../../lib/$(MACH64)/libzfs_core.so.1
1391 $(ROOT)/usr/lib/$(MACH64)/libfdisk.so.1:= \
1392 REALPATH=../../../../lib/$(MACH64)/libfdisk.so.1
1393 $(ROOT)/usr/lib/$(MACH64)/libfdisk.so:= \
1394 REALPATH=../../../../lib/$(MACH64)/libfdisk.so.1
1395 $(ROOT)/usr/lib/$(MACH64)/llib-ladm.ln:= \
1396 REALPATH=../../../../lib/$(MACH64)/llib-ladm.ln
1397 $(ROOT)/usr/lib/$(MACH64)/llib-laio.ln:= \
1398 REALPATH=../../../../lib/$(MACH64)/llib-laio.ln
1399 $(ROOT)/usr/lib/$(MACH64)/llib-lavl.ln:= \
1400 REALPATH=../../../../lib/$(MACH64)/llib-lavl.ln
1401 $(ROOT)/usr/lib/$(MACH64)/llib-lbsm.ln:= \
1402 REALPATH=../../../../lib/$(MACH64)/llib-lbsm.ln
1403 $(ROOT)/usr/lib/$(MACH64)/llib-lc.ln:= \
1404 REALPATH=../../../../lib/$(MACH64)/llib-lc.ln
1405 $(ROOT)/usr/lib/$(MACH64)/llib-lcmdutils.ln:= \
1406 REALPATH=../../../../lib/$(MACH64)/llib-lcmdutils.ln
1407 $(ROOT)/usr/lib/$(MACH64)/llib-lcontract.ln:= \
1408 REALPATH=../../../../lib/$(MACH64)/llib-lcontract.ln
1409 $(ROOT)/usr/lib/$(MACH64)/llib-lctf.ln:= \
1410 REALPATH=../../../../lib/$(MACH64)/llib-lctf.ln
1411 $(ROOT)/usr/lib/$(MACH64)/llib-lcurses.ln:= \
1412 REALPATH=../../../../lib/$(MACH64)/llib-lcurses.ln
1413 $(ROOT)/usr/lib/$(MACH64)/llib-ldevice.ln:= \
1414 REALPATH=../../../../lib/$(MACH64)/llib-ldevice.ln
1415 $(ROOT)/usr/lib/$(MACH64)/llib-ldevid.ln:= \
1416 REALPATH=../../../../lib/$(MACH64)/llib-ldevid.ln
1417 $(ROOT)/usr/lib/$(MACH64)/llib-ldevinfo.ln:= \
1418 REALPATH=../../../../lib/$(MACH64)/llib-ldevinfo.ln
1419 $(ROOT)/usr/lib/$(MACH64)/llib-ldhcutil.ln:= \
1420 REALPATH=../../../../lib/$(MACH64)/llib-ldhcutil.ln
1421 $(ROOT)/usr/lib/$(MACH64)/llib-ldl.ln:= \
1422 REALPATH=../../../../lib/$(MACH64)/llib-ldl.ln
1423 $(ROOT)/usr/lib/$(MACH64)/llib-ldoor.ln:= \
1424 REALPATH=../../../../lib/$(MACH64)/llib-ldoor.ln
1425 $(ROOT)/usr/lib/$(MACH64)/llib-lefi.ln:= \
1426 REALPATH=../../../../lib/$(MACH64)/llib-lefi.ln
1427 $(ROOT)/usr/lib/$(MACH64)/llib-lelf.ln:= \
1428 REALPATH=../../../../lib/$(MACH64)/llib-lelf.ln
1429 $(ROOT)/usr/lib/$(MACH64)/llib-lgen.ln:= \
1430 REALPATH=../../../../lib/$(MACH64)/llib-lgen.ln
1431 $(ROOT)/usr/lib/$(MACH64)/llib-linetutil.ln:= \
1432 REALPATH=../../../../lib/$(MACH64)/llib-linetutil.ln
1433 $(ROOT)/usr/lib/$(MACH64)/llib-lintl.ln:= \
1434 REALPATH=../../../../lib/$(MACH64)/llib-lintl.ln
1435 $(ROOT)/usr/lib/$(MACH64)/llib-lkstat.ln:= \
1436 REALPATH=../../../../lib/$(MACH64)/llib-lkstat.ln
1437 $(ROOT)/usr/lib/$(MACH64)/llib-lmd5.ln:= \
1438 REALPATH=../../../../lib/$(MACH64)/llib-lmd5.ln
1439 $(ROOT)/usr/lib/$(MACH64)/llib-lns1.ln:= \
1440 REALPATH=../../../../lib/$(MACH64)/llib-lns1.ln
1441 $(ROOT)/usr/lib/$(MACH64)/llib-lnvpair.ln:= \
1442 REALPATH=../../../../lib/$(MACH64)/llib-lnvpair.ln
1443 $(ROOT)/usr/lib/$(MACH64)/llib-lpam.ln:= \

```

```

1444 REALPATH=../../../../lib/$(MACH64)/llib-lpam.ln
1445 $(ROOT)/usr/lib/$(MACH64)/llib-lposix4.ln:= \
1446 REALPATH=../../../../lib/$(MACH64)/llib-lrt.ln
1447 $(ROOT)/usr/lib/$(MACH64)/llib-lpthread.ln:= \
1448 REALPATH=../../../../lib/$(MACH64)/llib-lpthread.ln
1449 $(ROOT)/usr/lib/$(MACH64)/llib-lresolv.ln:= \
1450 REALPATH=../../../../lib/$(MACH64)/llib-lresolv.ln
1451 $(ROOT)/usr/lib/$(MACH64)/llib-lrpcsvc.ln:= \
1452 REALPATH=../../../../lib/$(MACH64)/llib-lrpcsvc.ln
1453 $(ROOT)/usr/lib/$(MACH64)/llib-lrt.ln:= \
1454 REALPATH=../../../../lib/$(MACH64)/llib-lrt.ln
1455 $(ROOT)/usr/lib/$(MACH64)/llib-lrtld_db.ln:= \
1456 REALPATH=../../../../lib/$(MACH64)/llib-lrtld_db.ln
1457 $(ROOT)/usr/lib/$(MACH64)/llib-lscf.ln:= \
1458 REALPATH=../../../../lib/$(MACH64)/llib-lscf.ln
1459 $(ROOT)/usr/lib/$(MACH64)/llib-lsec.ln:= \
1460 REALPATH=../../../../lib/$(MACH64)/llib-lsec.ln
1461 $(ROOT)/usr/lib/$(MACH64)/llib-lsecdb.ln:= \
1462 REALPATH=../../../../lib/$(MACH64)/llib-lsecdb.ln
1463 $(ROOT)/usr/lib/$(MACH64)/llib-lsendfile.ln:= \
1464 REALPATH=../../../../lib/$(MACH64)/llib-lsendfile.ln
1465 $(ROOT)/usr/lib/$(MACH64)/llib-lsocket.ln:= \
1466 REALPATH=../../../../lib/$(MACH64)/llib-lsocket.ln
1467 $(ROOT)/usr/lib/$(MACH64)/llib-lsysevent.ln:= \
1468 REALPATH=../../../../lib/$(MACH64)/llib-lsysevent.ln
1469 $(ROOT)/usr/lib/$(MACH64)/llib-ltermcap.ln:= \
1470 REALPATH=../../../../lib/$(MACH64)/llib-ltermcap.ln
1471 $(ROOT)/usr/lib/$(MACH64)/llib-ltermplib.ln:= \
1472 REALPATH=../../../../lib/$(MACH64)/llib-lcurses.ln
1473 $(ROOT)/usr/lib/$(MACH64)/llib-lthread.ln:= \
1474 REALPATH=../../../../lib/$(MACH64)/llib-lthread.ln
1475 $(ROOT)/usr/lib/$(MACH64)/llib-lthread_db.ln:= \
1476 REALPATH=../../../../lib/$(MACH64)/llib-lc_db.ln
1477 $(ROOT)/usr/lib/$(MACH64)/llib-ltsnet.ln:= \
1478 REALPATH=../../../../lib/$(MACH64)/llib-ltsnet.ln
1479 $(ROOT)/usr/lib/$(MACH64)/llib-ltsol.ln:= \
1480 REALPATH=../../../../lib/$(MACH64)/llib-ltsol.ln
1481 $(ROOT)/usr/lib/$(MACH64)/llib-lumem.ln:= \
1482 REALPATH=../../../../lib/$(MACH64)/llib-lumem.ln
1483 $(ROOT)/usr/lib/$(MACH64)/llib-luuid.ln:= \
1484 REALPATH=../../../../lib/$(MACH64)/llib-luuid.ln
1485 $(ROOT)/usr/lib/$(MACH64)/llib-lxnet.ln:= \
1486 REALPATH=../../../../lib/$(MACH64)/llib-lxnet.ln
1487 $(ROOT)/usr/lib/$(MACH64)/llib-lzfs.ln:= \
1488 REALPATH=../../../../lib/$(MACH64)/llib-lzfs.ln
1489 $(ROOT)/usr/lib/$(MACH64)/llib-lzfs_core.ln:= \
1490 REALPATH=../../../../lib/$(MACH64)/llib-lzfs_core.ln
1491 $(ROOT)/usr/lib/$(MACH64)/llib-lfdisk.ln:= \
1492 REALPATH=../../../../lib/$(MACH64)/llib-lfdisk.ln
1493 $(ROOT)/usr/lib/$(MACH64)/nss_compat.so.1:= \
1494 REALPATH=../../../../lib/$(MACH64)/nss_compat.so.1
1495 $(ROOT)/usr/lib/$(MACH64)/nss_dns.so.1:= \
1496 REALPATH=../../../../lib/$(MACH64)/nss_dns.so.1
1497 $(ROOT)/usr/lib/$(MACH64)/nss_files.so.1:= \
1498 REALPATH=../../../../lib/$(MACH64)/nss_files.so.1
1499 $(ROOT)/usr/lib/$(MACH64)/nss_nis.so.1:= \
1500 REALPATH=../../../../lib/$(MACH64)/nss_nis.so.1
1501 $(ROOT)/usr/lib/$(MACH64)/nss_user.so.1:= \
1502 REALPATH=../../../../lib/$(MACH64)/nss_user.so.1
1503 $(ROOT)/usr/lib/fm/$(MACH64)/libfmevent.so.1:= \
1504 REALPATH=../../../../lib/fm/$(MACH64)/libfmevent.so.1
1505 $(ROOT)/usr/lib/fm/$(MACH64)/libfmevent.so:= \
1506 REALPATH=../../../../lib/fm/$(MACH64)/libfmevent.so.1
1507 $(ROOT)/usr/lib/fm/$(MACH64)/llib-lfmevent.ln:= \
1508 REALPATH=../../../../lib/fm/$(MACH64)/llib-lfmevent.ln

```

```

1510 i386_SYM.USRLIB= \
1511 /usr/lib/libfdisk.so \
1512 /usr/lib/libfdisk.so.1 \
1513 /usr/lib/llib-lfdisk \
1514 /usr/lib/llib-lfdisk.ln

1516 SYM.USRLIB= \
1517 $( $(MACH)_SYM.USRLIB) \
1518 /lib/libposix4.so \
1519 /lib/libposix4.so.1 \
1520 /lib/llib-lposix4 \
1521 /lib/llib-lposix4.ln \
1522 /lib/libthread_db.so \
1523 /lib/libthread_db.so.1 \
1524 /usr/lib/ld.so.1 \
1525 /usr/lib/libadm.so \
1526 /usr/lib/libadm.so.1 \
1527 /usr/lib/libaio.so \
1528 /usr/lib/libaio.so.1 \
1529 /usr/lib/libavl.so \
1530 /usr/lib/libavl.so.1 \
1531 /usr/lib/libbsm.so \
1532 /usr/lib/libbsm.so.1 \
1533 /usr/lib/libc.so \
1534 /usr/lib/libc.so.1 \
1535 /usr/lib/libc_db.so \
1536 /usr/lib/libc_db.so.1 \
1537 /usr/lib/libcmdutils.so \
1538 /usr/lib/libcmdutils.so.1 \
1539 /usr/lib/libcontract.so \
1540 /usr/lib/libcontract.so.1 \
1541 /usr/lib/libctf.so \
1542 /usr/lib/libctf.so.1 \
1543 /usr/lib/libcurses.so \
1544 /usr/lib/libcurses.so.1 \
1545 /usr/lib/libdevice.so \
1546 /usr/lib/libdevice.so.1 \
1547 /usr/lib/libdevvid.so \
1548 /usr/lib/libdevvid.so.1 \
1549 /usr/lib/libdevinfo.so \
1550 /usr/lib/libdevinfo.so.1 \
1551 /usr/lib/libdhcpagent.so \
1552 /usr/lib/libdhcpagent.so.1 \
1553 /usr/lib/libdhcputil.so \
1554 /usr/lib/libdhcputil.so.1 \
1555 /usr/lib/libdl.so \
1556 /usr/lib/libdl.so.1 \
1557 /usr/lib/libdmpi.so \
1558 /usr/lib/libdmpi.so.1 \
1559 /usr/lib/libdoor.so \
1560 /usr/lib/libdoor.so.1 \
1561 /usr/lib/libefi.so \
1562 /usr/lib/libefi.so.1 \
1563 /usr/lib/libelf.so \
1564 /usr/lib/libelf.so.1 \
1565 /usr/lib/libgen.so \
1566 /usr/lib/libgen.so.1 \
1567 /usr/lib/libinetutil.so \
1568 /usr/lib/libinetutil.so.1 \
1569 /usr/lib/libintl.so \
1570 /usr/lib/libintl.so.1 \
1571 /usr/lib/libkstat.so \
1572 /usr/lib/libkstat.so.1 \
1573 /usr/lib/liblddbg.so.4 \
1574 /usr/lib/libmd.so \
1575 /usr/lib/libmd.so.1 \

```

```

1576 /usr/lib/libmd5.so \
1577 /usr/lib/libmd5.so.1 \
1578 /usr/lib/libmeta.so \
1579 /usr/lib/libmeta.so.1 \
1580 /usr/lib/libmp.so \
1581 /usr/lib/libmp.so.1 \
1582 /usr/lib/libmp.so.2 \
1583 /usr/lib/libnsl.so \
1584 /usr/lib/libnsl.so.1 \
1585 /usr/lib/libnvpair.so \
1586 /usr/lib/libnvpair.so.1 \
1587 /usr/lib/libpam.so \
1588 /usr/lib/libpam.so.1 \
1589 /usr/lib/libposix4.so \
1590 /usr/lib/libposix4.so.1 \
1591 /usr/lib/libproc.so \
1592 /usr/lib/libproc.so.1 \
1593 /usr/lib/libpthread.so \
1594 /usr/lib/libpthread.so.1 \
1595 /usr/lib/librcm.so \
1596 /usr/lib/librcm.so.1 \
1597 /usr/lib/libresolv.so \
1598 /usr/lib/libresolv.so.1 \
1599 /usr/lib/libresolv.so.2 \
1600 /usr/lib/librestart.so \
1601 /usr/lib/librestart.so.1 \
1602 /usr/lib/librpcsvc.so \
1603 /usr/lib/librpcsvc.so.1 \
1604 /usr/lib/librt.so \
1605 /usr/lib/librt.so.1 \
1606 /usr/lib/librtld.so.1 \
1607 /usr/lib/librtld_db.so \
1608 /usr/lib/librtld_db.so.1 \
1609 /usr/lib/libscf.so \
1610 /usr/lib/libscf.so.1 \
1611 /usr/lib/libsec.so \
1612 /usr/lib/libsec.so.1 \
1613 /usr/lib/libsecdb.so \
1614 /usr/lib/libsecdb.so.1 \
1615 /usr/lib/libsendfile.so \
1616 /usr/lib/libsendfile.so.1 \
1617 /usr/lib/libsocket.so \
1618 /usr/lib/libsocket.so.1 \
1619 /usr/lib/libsysevent.so \
1620 /usr/lib/libsysevent.so.1 \
1621 /usr/lib/libtermcap.so \
1622 /usr/lib/libtermcap.so.1 \
1623 /usr/lib/libtermlib.so \
1624 /usr/lib/libtermlib.so.1 \
1625 /usr/lib/libthread.so \
1626 /usr/lib/libthread.so.1 \
1627 /usr/lib/libthread_db.so \
1628 /usr/lib/libthread_db.so.1 \
1629 /usr/lib/libtsnet.so \
1630 /usr/lib/libtsnet.so.1 \
1631 /usr/lib/libtsol.so \
1632 /usr/lib/libtsol.so.2 \
1633 /usr/lib/libumem.so \
1634 /usr/lib/libumem.so.1 \
1635 /usr/lib/libuuid.so \
1636 /usr/lib/libuuid.so.1 \
1637 /usr/lib/libuutil.so \
1638 /usr/lib/libuutil.so.1 \
1639 /usr/lib/libw.so \
1640 /usr/lib/libw.so.1 \
1641 /usr/lib/libxnet.so \

```

```

1642 /usr/lib/libxnet.so.1 \
1643 /usr/lib/libzfs.so \
1644 /usr/lib/libzfs.so.1 \
1645 /usr/lib/libzfs_core.so \
1646 /usr/lib/libzfs_core.so.1 \
1647 /usr/lib/libzfs_ladm \
1648 /usr/lib/libzfs_ladm.ln \
1649 /usr/lib/libzfs_laio \
1650 /usr/lib/libzfs_laio.ln \
1651 /usr/lib/libzfs_lavl \
1652 /usr/lib/libzfs_lavl.ln \
1653 /usr/lib/libzfs_lbsm \
1654 /usr/lib/libzfs_lbsm.ln \
1655 /usr/lib/libzfs_lc \
1656 /usr/lib/libzfs_lc.ln \
1657 /usr/lib/libzfs_lcmdutils \
1658 /usr/lib/libzfs_lcmdutils.ln \
1659 /usr/lib/libzfs_lcontract \
1660 /usr/lib/libzfs_lcontract.ln \
1661 /usr/lib/libzfs_lctf \
1662 /usr/lib/libzfs_lctf.ln \
1663 /usr/lib/libzfs_lcurls \
1664 /usr/lib/libzfs_lcurls.ln \
1665 /usr/lib/libzfs_ldevice \
1666 /usr/lib/libzfs_ldevice.ln \
1667 /usr/lib/libzfs_ldevid \
1668 /usr/lib/libzfs_ldevid.ln \
1669 /usr/lib/libzfs_ldevinfo \
1670 /usr/lib/libzfs_ldevinfo.ln \
1671 /usr/lib/libzfs_ldhcpagent \
1672 /usr/lib/libzfs_ldhcpagent.ln \
1673 /usr/lib/libzfs_ldhcputil \
1674 /usr/lib/libzfs_ldhcputil.ln \
1675 /usr/lib/libzfs_ldl \
1676 /usr/lib/libzfs_ldl.ln \
1677 /usr/lib/libzfs_ldoor \
1678 /usr/lib/libzfs_ldoor.ln \
1679 /usr/lib/libzfs_lefi \
1680 /usr/lib/libzfs_lefi.ln \
1681 /usr/lib/libzfs_elf \
1682 /usr/lib/libzfs_elf.ln \
1683 /usr/lib/libzfs_lgen \
1684 /usr/lib/libzfs_lgen.ln \
1685 /usr/lib/libzfs_linetutil \
1686 /usr/lib/libzfs_linetutil.ln \
1687 /usr/lib/libzfs_lintl \
1688 /usr/lib/libzfs_lintl.ln \
1689 /usr/lib/libzfs_lkstat \
1690 /usr/lib/libzfs_lkstat.ln \
1691 /usr/lib/libzfs_lmd5 \
1692 /usr/lib/libzfs_lmd5.ln \
1693 /usr/lib/libzfs_lmeta \
1694 /usr/lib/libzfs_lmeta.ln \
1695 /usr/lib/libzfs_lnsl \
1696 /usr/lib/libzfs_lnsl.ln \
1697 /usr/lib/libzfs_lnvpair \
1698 /usr/lib/libzfs_lnvpair.ln \
1699 /usr/lib/libzfs_lpam \
1700 /usr/lib/libzfs_lpam.ln \
1701 /usr/lib/libzfs_lposix4 \
1702 /usr/lib/libzfs_lposix4.ln \
1703 /usr/lib/libzfs_lpthread \
1704 /usr/lib/libzfs_lpthread.ln \
1705 /usr/lib/libzfs_lresolv \
1706 /usr/lib/libzfs_lresolv.ln \
1707 /usr/lib/libzfs_lrpcsvc \

```

```

1708 /usr/lib/llib-lrpcsvc.ln \
1709 /usr/lib/llib-lrt \
1710 /usr/lib/llib-lrt.ln \
1711 /usr/lib/llib-lrtld_db \
1712 /usr/lib/llib-lrtld_db.ln \
1713 /usr/lib/llib-lscf \
1714 /usr/lib/llib-lscf.ln \
1715 /usr/lib/llib-lsec \
1716 /usr/lib/llib-lsec.ln \
1717 /usr/lib/llib-lsecdb \
1718 /usr/lib/llib-lsecdb.ln \
1719 /usr/lib/llib-lsendfile \
1720 /usr/lib/llib-lsendfile.ln \
1721 /usr/lib/llib-lsocket \
1722 /usr/lib/llib-lsocket.ln \
1723 /usr/lib/llib-lsysevent \
1724 /usr/lib/llib-lsysevent.ln \
1725 /usr/lib/llib-ltermcap \
1726 /usr/lib/llib-ltermcap.ln \
1727 /usr/lib/llib-ltermlib \
1728 /usr/lib/llib-ltermlib.ln \
1729 /usr/lib/llib-lthread \
1730 /usr/lib/llib-lthread.ln \
1731 /usr/lib/llib-lthread_db \
1732 /usr/lib/llib-lthread_db.ln \
1733 /usr/lib/llib-ltsnet \
1734 /usr/lib/llib-ltsnet.ln \
1735 /usr/lib/llib-ltsol \
1736 /usr/lib/llib-ltsol.ln \
1737 /usr/lib/llib-lumem \
1738 /usr/lib/llib-lumem.ln \
1739 /usr/lib/llib-luuid \
1740 /usr/lib/llib-luuid.ln \
1741 /usr/lib/llib-lxnet \
1742 /usr/lib/llib-lxnet.ln \
1743 /usr/lib/llib-lzfs \
1744 /usr/lib/llib-lzfs.ln \
1745 /usr/lib/llib-lzfs_core \
1746 /usr/lib/llib-lzfs_core.ln \
1747 /usr/lib/nss_compat.so.1 \
1748 /usr/lib/nss_dns.so.1 \
1749 /usr/lib/nss_files.so.1 \
1750 /usr/lib/nss_nis.so.1 \
1751 /usr/lib/nss_user.so.1 \
1752 /usr/lib/fm/libfmevent.so \
1753 /usr/lib/fm/libfmevent.so.1 \
1754 /usr/lib/fm/llib-lfmevent \
1755 /usr/lib/fm/llib-lfmevent.ln

1757 sparcv9_SYM.USRLIB64=

1759 amd64_SYM.USRLIB64= \
1760 /usr/lib/amd64/libfdisk.so \
1761 /usr/lib/amd64/libfdisk.so.1 \
1762 /usr/lib/amd64/llib-lfdisk.ln

1765 SYM.USRLIB64= \
1766 $(MACH64)_SYM.USRLIB64 \
1767 /lib/$(MACH64)/libposix4.so \
1768 /lib/$(MACH64)/libposix4.so.1 \
1769 /lib/$(MACH64)/llib-lposix4.ln \
1770 /lib/$(MACH64)/libthread_db.so \
1771 /lib/$(MACH64)/libthread_db.so.1 \
1772 /usr/lib/$(MACH64)/ld.so.1 \
1773 /usr/lib/$(MACH64)/libadm.so \

```

```

1774 /usr/lib/$(MACH64)/libadm.so.1 \
1775 /usr/lib/$(MACH64)/libaio.so \
1776 /usr/lib/$(MACH64)/libaio.so.1 \
1777 /usr/lib/$(MACH64)/libavl.so \
1778 /usr/lib/$(MACH64)/libavl.so.1 \
1779 /usr/lib/$(MACH64)/libbsm.so \
1780 /usr/lib/$(MACH64)/libbsm.so.1 \
1781 /usr/lib/$(MACH64)/libc.so \
1782 /usr/lib/$(MACH64)/libc.so.1 \
1783 /usr/lib/$(MACH64)/libc_db.so \
1784 /usr/lib/$(MACH64)/libc_db.so.1 \
1785 /usr/lib/$(MACH64)/libcmdutils.so \
1786 /usr/lib/$(MACH64)/libcmdutils.so.1 \
1787 /usr/lib/$(MACH64)/libcontract.so \
1788 /usr/lib/$(MACH64)/libcontract.so.1 \
1789 /usr/lib/$(MACH64)/libctf.so \
1790 /usr/lib/$(MACH64)/libctf.so.1 \
1791 /usr/lib/$(MACH64)/libcurses.so \
1792 /usr/lib/$(MACH64)/libcurses.so.1 \
1793 /usr/lib/$(MACH64)/libdevice.so \
1794 /usr/lib/$(MACH64)/libdevice.so.1 \
1795 /usr/lib/$(MACH64)/libdeviceid.so \
1796 /usr/lib/$(MACH64)/libdeviceid.so.1 \
1797 /usr/lib/$(MACH64)/libdevinfo.so \
1798 /usr/lib/$(MACH64)/libdevinfo.so.1 \
1799 /usr/lib/$(MACH64)/libdhcputil.so \
1800 /usr/lib/$(MACH64)/libdhcputil.so.1 \
1801 /usr/lib/$(MACH64)/libdl.so \
1802 /usr/lib/$(MACH64)/libdl.so.1 \
1803 /usr/lib/$(MACH64)/libdlpi.so \
1804 /usr/lib/$(MACH64)/libdlpi.so.1 \
1805 /usr/lib/$(MACH64)/libdoor.so \
1806 /usr/lib/$(MACH64)/libdoor.so.1 \
1807 /usr/lib/$(MACH64)/libefi.so \
1808 /usr/lib/$(MACH64)/libefi.so.1 \
1809 /usr/lib/$(MACH64)/libelf.so \
1810 /usr/lib/$(MACH64)/libelf.so.1 \
1811 /usr/lib/$(MACH64)/libgen.so \
1812 /usr/lib/$(MACH64)/libgen.so.1 \
1813 /usr/lib/$(MACH64)/libinetutil.so \
1814 /usr/lib/$(MACH64)/libinetutil.so.1 \
1815 /usr/lib/$(MACH64)/libintl.so \
1816 /usr/lib/$(MACH64)/libintl.so.1 \
1817 /usr/lib/$(MACH64)/libkstat.so \
1818 /usr/lib/$(MACH64)/libkstat.so.1 \
1819 /usr/lib/$(MACH64)/liblddbg.so.4 \
1820 /usr/lib/$(MACH64)/libmd.so \
1821 /usr/lib/$(MACH64)/libmd.so.1 \
1822 /usr/lib/$(MACH64)/libmd5.so \
1823 /usr/lib/$(MACH64)/libmd5.so.1 \
1824 /usr/lib/$(MACH64)/libmp.so \
1825 /usr/lib/$(MACH64)/libmp.so.2 \
1826 /usr/lib/$(MACH64)/libnsl.so \
1827 /usr/lib/$(MACH64)/libnsl.so.1 \
1828 /usr/lib/$(MACH64)/libnvpair.so \
1829 /usr/lib/$(MACH64)/libnvpair.so.1 \
1830 /usr/lib/$(MACH64)/libpam.so \
1831 /usr/lib/$(MACH64)/libpam.so.1 \
1832 /usr/lib/$(MACH64)/libposix4.so \
1833 /usr/lib/$(MACH64)/libposix4.so.1 \
1834 /usr/lib/$(MACH64)/libproc.so \
1835 /usr/lib/$(MACH64)/libproc.so.1 \
1836 /usr/lib/$(MACH64)/libpthread.so \
1837 /usr/lib/$(MACH64)/libpthread.so.1 \
1838 /usr/lib/$(MACH64)/librcm.so \
1839 /usr/lib/$(MACH64)/librcm.so.1 \

```

```

1840 /usr/lib/$(MACH64)/libresolv.so \
1841 /usr/lib/$(MACH64)/libresolv.so.2 \
1842 /usr/lib/$(MACH64)/librestart.so \
1843 /usr/lib/$(MACH64)/librestart.so.1 \
1844 /usr/lib/$(MACH64)/librpcsvc.so \
1845 /usr/lib/$(MACH64)/librpcsvc.so.1 \
1846 /usr/lib/$(MACH64)/librt.so \
1847 /usr/lib/$(MACH64)/librt.so.1 \
1848 /usr/lib/$(MACH64)/librtld.so.1 \
1849 /usr/lib/$(MACH64)/librtld_db.so \
1850 /usr/lib/$(MACH64)/librtld_db.so.1 \
1851 /usr/lib/$(MACH64)/libscf.so \
1852 /usr/lib/$(MACH64)/libscf.so.1 \
1853 /usr/lib/$(MACH64)/libsec.so \
1854 /usr/lib/$(MACH64)/libsec.so.1 \
1855 /usr/lib/$(MACH64)/libsecdb.so \
1856 /usr/lib/$(MACH64)/libsecdb.so.1 \
1857 /usr/lib/$(MACH64)/libsendfile.so \
1858 /usr/lib/$(MACH64)/libsendfile.so.1 \
1859 /usr/lib/$(MACH64)/libsocket.so \
1860 /usr/lib/$(MACH64)/libsocket.so.1 \
1861 /usr/lib/$(MACH64)/libsysevent.so \
1862 /usr/lib/$(MACH64)/libsysevent.so.1 \
1863 /usr/lib/$(MACH64)/libtermcap.so \
1864 /usr/lib/$(MACH64)/libtermcap.so.1 \
1865 /usr/lib/$(MACH64)/libtermmlib.so \
1866 /usr/lib/$(MACH64)/libtermmlib.so.1 \
1867 /usr/lib/$(MACH64)/libthread.so \
1868 /usr/lib/$(MACH64)/libthread.so.1 \
1869 /usr/lib/$(MACH64)/libthread_db.so \
1870 /usr/lib/$(MACH64)/libthread_db.so.1 \
1871 /usr/lib/$(MACH64)/libtsnet.so \
1872 /usr/lib/$(MACH64)/libtsnet.so.1 \
1873 /usr/lib/$(MACH64)/libtsol.so \
1874 /usr/lib/$(MACH64)/libtsol.so.2 \
1875 /usr/lib/$(MACH64)/libumem.so \
1876 /usr/lib/$(MACH64)/libumem.so.1 \
1877 /usr/lib/$(MACH64)/libuuid.so \
1878 /usr/lib/$(MACH64)/libuuid.so.1 \
1879 /usr/lib/$(MACH64)/libutil.so \
1880 /usr/lib/$(MACH64)/libutil.so.1 \
1881 /usr/lib/$(MACH64)/libw.so \
1882 /usr/lib/$(MACH64)/libw.so.1 \
1883 /usr/lib/$(MACH64)/libxnet.so \
1884 /usr/lib/$(MACH64)/libxnet.so.1 \
1885 /usr/lib/$(MACH64)/libzfs.so \
1886 /usr/lib/$(MACH64)/libzfs.so.1 \
1887 /usr/lib/$(MACH64)/libzfs_core.so \
1888 /usr/lib/$(MACH64)/libzfs_core.so.1 \
1889 /usr/lib/$(MACH64)/llib-ladm.ln \
1890 /usr/lib/$(MACH64)/llib-laio.ln \
1891 /usr/lib/$(MACH64)/llib-lavl.ln \
1892 /usr/lib/$(MACH64)/llib-lbsm.ln \
1893 /usr/lib/$(MACH64)/llib-lc.ln \
1894 /usr/lib/$(MACH64)/llib-lcmdutils.ln \
1895 /usr/lib/$(MACH64)/llib-lcontract.ln \
1896 /usr/lib/$(MACH64)/llib-lctf.ln \
1897 /usr/lib/$(MACH64)/llib-lcurses.ln \
1898 /usr/lib/$(MACH64)/llib-ldevice.ln \
1899 /usr/lib/$(MACH64)/llib-ldevinfo.ln \
1900 /usr/lib/$(MACH64)/llib-ldevinfo.ln \
1901 /usr/lib/$(MACH64)/llib-ldhcputil.ln \
1902 /usr/lib/$(MACH64)/llib-ldl.ln \
1903 /usr/lib/$(MACH64)/llib-ldoor.ln \
1904 /usr/lib/$(MACH64)/llib-lefi.ln \
1905 /usr/lib/$(MACH64)/llib-lelf.ln \

```

```

1906 /usr/lib/$(MACH64)/llib-lgen.ln \
1907 /usr/lib/$(MACH64)/llib-linetutil.ln \
1908 /usr/lib/$(MACH64)/llib-lint1.ln \
1909 /usr/lib/$(MACH64)/llib-lkstat.ln \
1910 /usr/lib/$(MACH64)/llib-lmd5.ln \
1911 /usr/lib/$(MACH64)/llib-lnsl.ln \
1912 /usr/lib/$(MACH64)/llib-lnvpair.ln \
1913 /usr/lib/$(MACH64)/llib-lpam.ln \
1914 /usr/lib/$(MACH64)/llib-lposix4.ln \
1915 /usr/lib/$(MACH64)/llib-lpthread.ln \
1916 /usr/lib/$(MACH64)/llib-lresolv.ln \
1917 /usr/lib/$(MACH64)/llib-lrpcsvc.ln \
1918 /usr/lib/$(MACH64)/llib-lrt.ln \
1919 /usr/lib/$(MACH64)/llib-lrtld_db.ln \
1920 /usr/lib/$(MACH64)/llib-lscf.ln \
1921 /usr/lib/$(MACH64)/llib-lsec.ln \
1922 /usr/lib/$(MACH64)/llib-lsecdb.ln \
1923 /usr/lib/$(MACH64)/llib-lsendfile.ln \
1924 /usr/lib/$(MACH64)/llib-lsocket.ln \
1925 /usr/lib/$(MACH64)/llib-lsysevent.ln \
1926 /usr/lib/$(MACH64)/llib-ltermcap.ln \
1927 /usr/lib/$(MACH64)/llib-ltermmlib.ln \
1928 /usr/lib/$(MACH64)/llib-lthread.ln \
1929 /usr/lib/$(MACH64)/llib-lthread_db.ln \
1930 /usr/lib/$(MACH64)/llib-ltsnet.ln \
1931 /usr/lib/$(MACH64)/llib-ltsol.ln \
1932 /usr/lib/$(MACH64)/llib-lumem.ln \
1933 /usr/lib/$(MACH64)/llib-luuid.ln \
1934 /usr/lib/$(MACH64)/llib-lxnet.ln \
1935 /usr/lib/$(MACH64)/llib-lzfs.ln \
1936 /usr/lib/$(MACH64)/llib-lzfs_core.ln \
1937 /usr/lib/$(MACH64)/nss_compat.so.1 \
1938 /usr/lib/$(MACH64)/nss_dns.so.1 \
1939 /usr/lib/$(MACH64)/nss_files.so.1 \
1940 /usr/lib/$(MACH64)/nss_nis.so.1 \
1941 /usr/lib/$(MACH64)/nss_user.so.1 \
1942 /usr/lib/fm/$(MACH64)/libfmevent.so \
1943 /usr/lib/fm/$(MACH64)/libfmevent.so.1 \
1944 /usr/lib/fm/$(MACH64)/llib-lfmevent.ln

1946 #
1947 # usr/src/Makefile uses INS.dir for any member of ROOTDIRS, the fact
1948 # these are symlinks to files has no bearing on this.
1949 #
1950 $(FILELINKS:%=$(ROOT)%):= \
1951     INS.dir= -$(RM) $@; $(SYMLINK) $(REALPATH) $@

```

new/usr/src/cmd/oamuser/group/Makefile

1

```
*****
2042 Mon Sep  9 17:55:48 2013
new/usr/src/cmd/oamuser/group/Makefile
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
22 # Use is subject to license terms.
23 #
24 # Copyright (c) 2013 RackTop Systems.
25 #
26 #endif /* ! codereview */
27 # cmd/oamuser/group/Makefile
28 #

30 include ../../Makefile.cmd

32 GROUPADD=      groupadd
33 GROUPDEL=      groupdel
34 GROUPMOD=      groupmod

36 SBINPROG=      $(GROUPADD)  $(GROUPDEL)  $(GROUPMOD)

38 PROG=          $(SBINPROG)

40 ADD_OBJ=       groupadd.o  add_group.o  messages.o
24 ADD_OBJ=       groupadd.o  add_group.o  gid.o        messages.o
41 DEL_OBJ=       groupdel.o  del_group.o  messages.o
42 MOD_OBJ=       groupmod.o  mod_group.o  messages.o

44 OBJECTS=       $(ADD_OBJ)   $(DEL_OBJ)   $(MOD_OBJ)
45 SRCS=          $(OBJECTS:.o=.c)

47 LIBDIR=        ../lib
48 LIBUSRGRP=     $(LIBDIR)/lib.a
49 LOCAL=         ../inc
50 HERE=          .
51 LINTFLAGS=     -u

53 INSSBINPROG=   $(SBINPROG:%=$(ROOTUSRSBIN)/%)

55 CPPFLAGS=      -I$(HERE) -I$(LOCAL) $(CPPFLAGS.master)
40 CERRWARN +=   -_gcc=-Wno-type-limits
```

new/usr/src/cmd/oamuser/group/Makefile

2

```
56 FILEMODE=     0555

58 $(GROUPADD) := OBJS = $(ADD_OBJ)
59 $(GROUPADD) := LDLIBS += $(LIBUSRGRP) -lcmdutils
44 $(GROUPADD) := LDLIBS += $(LIBUSRGRP)

61 $(GROUPDEL) := OBJS = $(DEL_OBJ)
62 $(GROUPDEL) := LDLIBS += $(LIBUSRGRP)

64 $(GROUPMOD) := OBJS = $(MOD_OBJ)
65 $(GROUPMOD) := LDLIBS += $(LIBUSRGRP)

67 all:          $(PROG)

69 $(PROG):      $$$(OBJS) $(LIBUSRGRP)
70               $(LINK.c) $(OBJS) -o $$@ $(LDLIBS)
71               $(POST_PROCESS)

73 $(GROUPADD):  $(ADD_OBJ)
74 $(GROUPMOD):  $(MOD_OBJ)
75 $(GROUPDEL):  $(DEL_OBJ)

77 install:     all $(INSSBINPROG)

79 clean:
80              $(RM) $(OBJECTS)

82 lint:        lint_SRCS

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



```

*****
2763 Mon Sep  9 17:55:49 2013
new/usr/src/lib/libcndutils/common/gid.c
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22  * Copyright 2007 Sun Microsystems, Inc. All rights reserved.
23  * Use is subject to license terms.
24 */

26 #endif /* ! codereview */
27 /*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
28 /*      All Rights Reserved      */

30 /*
31  * Copyright (c) 2013 RackTop Systems.
32  */
33 #endif /* ! codereview */

35 #include <errno.h>
25 #pragma ident "%Z%M% %I% %E% SMI" /* SVr4.0 1.5 */

36 #include <sys/types.h>
37 #include <stdio.h>
38 #include <userdefs.h>
39 #include <grp.h>
40 #include <libcndutils.h>
41 #endif /* ! codereview */

43 static int findunusedgid(gid_t start, gid_t stop, gid_t *ret);
44 static boolean_t isreservedgid(gid_t gid);
30 #include <sys/param.h>
31 #ifndef MAXUID
32 #include <limits.h>
33 #ifdef UID_MAX
34 #define MAXUID UID_MAX
35 #else
36 #define MAXUID 60000
37 #endif
38 #endif

46 /*

```

```

47  * Find the highest unused uid. If the highest unused gid is "stop",
48  * then attempt to find a hole in the range. Returns 0 on success.
49  */
50 int
51 findnextgid(gid_t start, gid_t stop, gid_t *ret)
52 {
53     gid_t gid = start;
54     struct group *grp;
55     boolean_t overflow = B_FALSE;

57     setgrent();
58     for (grp = getgrent(); grp != NULL; grp = getgrent()) {
59         if (isreservedgid(grp->gr_gid)) /* Skip reserved IDs */
60             continue;
61         if (grp->gr_gid >= gid) {
62             if (grp->gr_gid == stop) { /* Overflow check */
63                 overflow = B_TRUE;
64                 break;
65             }
66             gid = grp->gr_gid + 1;
67         }
68     }
69     if (grp == NULL && errno != 0) {
70         endgrent();
71         return (-1);
72     }
73     endgrent();
74     if (overflow == B_TRUE) /* Find a hole */
75         return (findunusedgid(start, stop, ret));
76     while (isreservedgid(gid) && gid < stop) /* Skip reserved IDs */
77         gid++;
78     *ret = gid;
79     return (0);
80 }

82 /*
83  * Check to see whether the gid is a reserved gid
84  * Check to see that the gid is not a reserved gid
85  * -- nobody, noaccess or nogroup
86  */
86 static boolean_t
87 isreservedgid(gid_t gid)
88 static int
89 isvalidgid(gid_t gid)
88 {
89     return (gid == 60001 || gid == 60002 || gid == 65534);
47     return (gid != 60001 && gid != 60002 && gid != 65534);
90 }

92 /*
93  * findunusedgid() attempts to return the next valid usable id between the
94  * supplied upper and lower limits. Returns 0 on success.
95  */
96 static int
97 findunusedgid(gid_t start, gid_t stop, gid_t *ret)
50 gid_t
51 findnextgid()
98 {
53     FILE *fptr;
54     gid_t last, next;
99     gid_t gid;

101     for (gid = start; gid <= stop; gid++) {
102         if (isreservedgid(gid))
103             continue;
104         if (getgrgid(gid) == NULL) {

```

```

105         if (errno != 0)
106             return (-1);
107
108     if (fscanf(fptr, "%u\n", &next) == EOF) {
109         (void) pclose(fptr);
110         return (DEFRID + 1);
111     }
112
113     /*
114     * Sort the used GIDs in decreasing order to return MAXUSED + 1
115     */
116     if ((fptr = popen("exec sh -c "
117         "\"getent group/cut -f3 -d:/sort -nr/uniq \" 2>/dev/null",
118         "r")) == NULL)
119         return (-1);
120
121     if (fscanf(fptr, "%u\n", &next) == EOF) {
122         (void) pclose(fptr);
123         return (DEFRID + 1);
124     }
125
126     /*
127     * 'next' is now the highest allocated gid.
128     * The simplest allocation is where we just add one, and obtain
129     * a valid gid. If this fails look for a hole in the gid range ..
130     */
131
132     last = MAXUID;          /* upper limit */
133     gid = -1;              /* start invalid */
134     do {
135         if (!isvalidgid(next))
136             continue;
137
138         if (next <= DEFRID) {
139             if (last != DEFRID + 1)
140                 gid = DEFRID + 1;
141             break;
142         }
143
144         if ((gid = next + 1) != last) {
145             while (!isvalidgid(gid))
146                 gid++;
147             if (gid > 0 && gid < last)
148                 break;
149         }
150     }
151     if (gid > stop)
152         return (-1);
153     *ret = gid;
154     return (0);
155
156     gid = -1;
157     last = next;
158
159     } while (fscanf(fptr, "%u\n", &next) != EOF);
160
161     (void) pclose(fptr);
162
163     return (gid);
164 }
165
166 unchanged portion omitted

```

new/usr/src/cmd/oamuser/group/groupadd.c

1

```
*****
3290 Mon Sep  9 17:55:49 2013
new/usr/src/cmd/oamuser/group/groupadd.c
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
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 2005 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25 */

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

30 /*
31  * Copyright (c) 2013 RackTop Systems.
32  */

33 #pragma ident      "%Z%M% %I%      %E% SMI"

34 #include      <sys/types.h>
35 #include      <sys/param.h>
36 #endif /* ! codereview */
37 #include      <stdio.h>
38 #include      <stdlib.h>
39 #include      <ctype.h>
40 #include      <limits.h>
41 #include      <userdefs.h>
42 #include      <users.h>
43 #include      <errno.h>
44 #include      <libcmdutils.h>
45 #endif /* ! codereview */
46 #include      "messages.h"

48 extern int  errmsg();
34 extern gid_t  findnextgid();
49 extern int  valid_gid(), add_group();

51 /*
52  * groupadd [-g gid [-o]] group
53  *
54  *      This command adds new groups to the system.  Arguments are:
```

new/usr/src/cmd/oamuser/group/groupadd.c

2

```
55 *
56 *      gid - a gid_t less than MAXUID
57 *      group - a string of printable characters excluding colon(:) and less
58 *              than MAXGLEN characters long.
59 */

61 char *cmdname = "groupadd";

63 int
64 main(int argc, char *argv[])
65 {
66     int ch;                                /* return from getopt */
67     gid_t gid;                              /* group id */
68     int oflag = 0; /* flags */
69     int rc;
70     char *gidstr = NULL; /* gid from command line */
71     char *grpname; /* group name from command line */
72     int warning;

74     while ((ch = getopt(argc, argv, "g:o")) != EOF)
75     {
76         switch (ch) {
77             case 'g':
78                 gidstr = optarg;
79                 break;
80             case 'o':
81                 oflag++;
82                 break;
83             case '?':
84                 errmsg(M_AUSAGE);
85                 exit(EX_SYNTAX);
86         }
87     }

88     if ((oflag && !gidstr) || optind != argc - 1) {
89         errmsg(M_AUSAGE);
90         exit(EX_SYNTAX);
91     }

92     grpname = argv[optind];

94     switch (valid_gname(grpname, NULL, &warning)) {
95     case INVALID:
96         errmsg(M_GRP_INVALID, grpname);
97         exit(EX_BADARG);
98         /*NOTREACHED*/
99     case NOTUNIQUE:
100         errmsg(M_GRP_USED, grpname);
101         exit(EX_NAME_EXISTS);
102         /*NOTREACHED*/
103     }
104     if (warning)
105         warningmsg(warning, grpname);

107     if (gidstr) {
108         /* Given a gid string - validate it */
109         char *ptr;

111         errno = 0;
112         gid = (gid_t)strtol(gidstr, &ptr, 10);

114         if (*ptr || errno == ERANGE) {
115             errmsg(M_GID_INVALID, gidstr);
116             exit(EX_BADARG);
117         }

119         switch (valid_gid(gid, NULL)) {
120             case RESERVED:
```

```
121             errmsg(M_RESERVED, gid);
122             break;
123
124         case NOTUNIQUE:
125             if (!oflag) {
126                 errmsg(M_GRP_USED, gidstr);
127                 exit(EX_ID_EXISTS);
128             }
129             break;
130
131         case INVALID:
132             errmsg(M_GID_INVALID, gidstr);
133             exit(EX_BADARG);
134
135         case TOOBIG:
136             errmsg(M_TOOBIG, gid);
137             exit(EX_BADARG);
138     }
139
140     } else {
141
142         if (findnextgid(DEFRID+1, MAXUID, &gid) != 0) {
143             if ((gid = findnextgid()) < 0) {
144                 errmsg(M_GID_INVALID, "default id");
145                 exit(EX_ID_EXISTS);
146             }
147         }
148     }
149
150     if ((rc = add_group(grpname, gid)) != EX_SUCCESS)
151         errmsg(M_UPDATE, "created");
152
153     return (rc);
154 }
155
156 unchanged_portion_omitted
```

```

*****
3096 Mon Sep  9 17:55:50 2013
new/usr/src/cmd/oamuser/user/Makefile
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
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) 1990, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright (c) 2013 RackTop Systems.
24 #endif /* ! codereview */
25 #
26 # cmd/oamuser/user/Makefile
27 #

29 include ../../Makefile.cmd

31 GREP=      grep

33 USERADD=   useradd
34 USERDEL=   userdel
35 USERMOD=   usermod
36 ROLEADD=   roleadd
37 ROLEDEL=   roledel
38 ROLEMOD=   rolemod

40 SBINPROG=  $(USERADD) $(USERDEL) $(USERMOD)
41 #
42 #      Removing sysadm: deleted $(SYSADMPROG) from this target.
43 #
44 PROG=      $(SBINPROG)
45 PRODUCT=  $(PROG)

47 ADD_OBJ=  useradd.o  homedir.o  groups.o  call_pass.o \
48            userdefs.o messages.o  val_lgrp.o funcs.o \
49            val_lprj.o proj.o
23 ADD_OBJ=  useradd.o  uid.o      homedir.o \
24            groups.o  call_pass.o userdefs.o messages.o \
25            val_lgrp.o funcs.o    val_lprj.o proj.o

51 DEL_OBJ=  userdel.o  call_pass.o  rmfiles.o  isbusy.o \
52            groups.o  messages.o    funcs.o    proj.o

54 MOD_OBJ=  usermod.o  movedir.o   groups.o   rmfiles.o \

```

```

55          call_pass.o  isbusy.o  homedir.o  userdefs.o \
56          messages.o  val_lgrp.o  funcs.o    val_lprj.o \
57          proj.o
30 MOD_OBJ= usermod.o  uid.o      movedir.o  groups.o \
31          rmfiles.o  call_pass.o  isbusy.o  homedir.o \
32          userdefs.o messages.o  val_lgrp.o funcs.o \
33          val_lprj.o proj.o

59 OBJECTS=  $(ADD_OBJ)  $(DEL_OBJ)  $(MOD_OBJ)

61 SRCS=      $(OBJECTS:.o=.c)

63 LIBDIR=    ../lib
64 LIBUSRGRP= $(LIBDIR)/lib.a
65 LIBADM=    -ladm
66 LOCAL=     ../inc
67 HERE=     .
68 LINTFLAGS= -u

70 ROOTSKEL=  $(ROOTETC)/skel
71 INSSBINPROG=$(SBINPROG:%=$(ROOTUSRSBIN)/%)
72 INSSKELFILE=$(SKELFILE:%=$(ROOTSKEL)/%)

74 CPPFLAGS=  -I$(HERE) -I$(LOCAL) $(CPPFLAGS.master)
75 CERRWARN += -_gcc=-Wno-implicit-function-declaration
52 CERRWARN += -_gcc=-Wno-type-limits
53 CERRWARN += -_gcc=-Wno-uninitialized
54 CERRWARN += -_gcc=-Wno-parentheses

77 $(INSSBINPROG) := FILEMODE = 0555
78 $(INSSYSADMPROG) := FILEMODE = 0500
79 $(INSSKELFILE) := FILEMODE = 0644

81 $(USERADD) := OBJS = $(ADD_OBJ)
82 $(USERADD) := LIBS = $(LIBUSRGRP)
83 $(USERADD) := LDLIBS += -lcmdutils
84 #endif /* ! codereview */

86 $(USERDEL) := OBJS = $(DEL_OBJ)
87 $(USERDEL) := LIBS = $(LIBUSRGRP)

89 $(USERMOD) := OBJS = $(MOD_OBJ)
90 $(USERMOD) := LIBS = $(LIBUSRGRP)

92 LDLIBS +=  -lbsm -lnsl -lsecdb -lproject -ltsol

94 .PARALLEL: $(OBJECTS)

96 all:       $(PRODUCT)

98 $(PROG):   $$$(OBJS) $$$(LIBS)
99            $(LINK.c) $(OBJS) -o @$ $(LIBS) $(LDLIBS)
100           $(POST_PROCESS)

102 $(USERADD): $(ADD_OBJ)
103 $(USERMOD): $(MOD_OBJ)
104 $(USERDEL): $(DEL_OBJ)

106 install:  all .WAIT $(ROOTSKEL) $(INSSBINPROG) $(INSSKELFILE)
107            $(RM) $(ROOTUSRSBIN)/$(ROLEADD)
108            $(LN) $(ROOTUSRSBIN)/$(USERADD) $(ROOTUSRSBIN)/$(ROLEADD)
109            $(RM) $(ROOTUSRSBIN)/$(ROLEDEL)
110            $(LN) $(ROOTUSRSBIN)/$(USERDEL) $(ROOTUSRSBIN)/$(ROLEDEL)
111            $(RM) $(ROOTUSRSBIN)/$(ROLEMOD)
112            $(LN) $(ROOTUSRSBIN)/$(USERMOD) $(ROOTUSRSBIN)/$(ROLEMOD)

```

```
114 clean:
115     $(RM) $(OBJECTS)

117 lint:  lint_SRCS

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

```

*****
11350 Mon Sep  9 17:55:50 2013
new/usr/src/cmd/oamuser/user/funcs.c
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
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) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
23  * Copyright (c) 2013 RackTop Systems.
24 #endif /* ! codereview */
25 */

27 #include <stdio.h>
28 #include <stdlib.h>
29 #include <strings.h>
30 #include <auth_attr.h>
31 #include <prof_attr.h>
32 #include <user_attr.h>
33 #include <project.h>
34 #include <secdb.h>
35 #include <pwd.h>
36 #include <unistd.h>
37 #include <priv.h>
38 #include <errno.h>
39 #include <ctype.h>
40 #include <nss.h>
41 #include <bsm/libbsm.h>
42 #include <tsol/label.h>
43 #include "funcs.h"
44 #include "messages.h"
45 #undef GROUP
46 #include "userdefs.h"

48 typedef struct ua_key {
49     const char    *key;
50     const char    *(*check)(const char *);
51     const char    *errstr;
52     char          *newvalue;
53 } ua_key_t;

55 static const char role[] = "role name";
56 static const char prof[] = "profile name";
57 static const char proj[] = "project name";

```

```

58 static const char priv[] = "privilege set";
59 static const char auth[] = "authorization";
60 static const char type[] = "user type";
61 static const char lock[] = "lock_after_retries value";
62 static const char label[] = "label";
63 static const char idlecmd[] = "idlecmd value";
64 static const char idletime[] = "idletime value";
65 static const char auditflags[] = "audit mask";
66 static char      auditerr[256];

69 static const char *check_auth(const char *);
70 static const char *check_prof(const char *);
71 static const char *check_role(const char *);
72 static const char *check_proj(const char *);
73 static const char *check_privset(const char *);
74 static const char *check_type(const char *);
75 static const char *check_lock_after_retries(const char *);
76 static const char *check_label(const char *);
77 static const char *check_idlecmd(const char *);
78 static const char *check_idletime(const char *);
79 static const char *check_auditflags(const char *);

81 int nkeys;

83 static ua_key_t keys[] = {
84     /* First entry is always set correctly in main() */
85     { USERATTR_TYPE_KW,    check_type,    type },
86     { USERATTR_AUTHS_KW,  check_auth,    auth },
87     { USERATTR_PROFILES_KW, check_prof,    prof },
88     { USERATTR_ROLES_KW,  check_role,    role },
89     { USERATTR_DEFAULTPROJ_KW, check_proj,    proj },
90     { USERATTR_LIMPRIV_KW, check_privset, priv },
91     { USERATTR_DFLTPRIV_KW, check_privset, priv },
92     { USERATTR_LOCK_AFTER_RETRIES_KW, check_lock_after_retries, lock },
93     { USERATTR_CLEARANCE, check_label,    label },
94     { USERATTR_MINLABEL,  check_label,    label },
95     { USERATTR_IDLECMD_KW, check_idlecmd, idlecmd },
96     { USERATTR_IDLETIME_KW, check_idletime, idletime },
97     { USERATTR_AUDIT_FLAGS_KW, check_auditflags, auditflags },
98 };

100 #define NKEYS    (sizeof (keys)/sizeof (ua_key_t))

102 /*
103  * Change a key, there are three different call sequences:
104  *
105  *           key, value    - key with option letter, value.
106  *           NULL, value   - -K key=value option.
107  */

109 void
110 change_key(const char *key, char *value)
111 {
112     int i;
113     const char *res;

115     if (key == NULL) {
116         key = value;
117         value = strchr(value, '=');
118         /* Bad value */
119         if (value == NULL) {
120             errmsg(M_INVALID_VALUE);
121             exit(EX_BADARG);
122         }
123         *value++ = '\0';

```

```

124     }
125
126     for (i = 0; i < NKEYS; i++) {
127         if (strcmp(key, keys[i].key) == 0) {
128             if (keys[i].newvalue != NULL) {
129                 /* Can't set a value twice */
130                 errmsg(M_REDEFINED_KEY, key);
131                 exit(EX_BADARG);
132             }
133
134             if (keys[i].check != NULL &&
135                 (res = keys[i].check(value)) != NULL) {
136                 errmsg(M_INVALID, res, keys[i].errstr);
137                 exit(EX_BADARG);
138             }
139             keys[i].newvalue = value;
140             nkeys++;
141             return;
142         }
143     }
144     errmsg(M_INVALID_KEY, key);
145     exit(EX_BADARG);
146 }
147
148 /*
149  * Add the keys to the argument vector.
150  */
151 void
152 addkey_args(char **argv, int *index)
153 {
154     int i;
155
156     for (i = 0; i < NKEYS; i++) {
157         const char *key = keys[i].key;
158         char *val = keys[i].newvalue;
159         size_t len;
160         char *arg;
161
162         if (val == NULL)
163             continue;
164
165         len = strlen(key) + strlen(val) + 2;
166         arg = malloc(len);
167
168         (void) snprintf(arg, len, "%s=%s", key, val);
169         argv[*index++] = "-K";
170         argv[*index++] = arg;
171     }
172 }
173
174 /*
175  * Propose a default value for a key and get the actual value back.
176  * If the proposed default value is NULL, return the actual value set.
177  * The key argument is the user_attr key.
178  */
179 char *
180 getsetdefval(const char *key, char *dflt)
181 {
182     int i;
183
184     for (i = 0; i < NKEYS; i++)
185         if (strcmp(keys[i].key, key) == 0) {
186             if (strcmp(keys[i].key, key) == 0)
187                 if (keys[i].newvalue != NULL)
188                     return (keys[i].newvalue);
189             else

```

```

189         }
190     }
191     return (keys[i].newvalue = dflt);
192 }
193
194
195 char *
196 getusertype(char *cmdname)
197 {
198     static char usertype[MAX_TYPE_LENGTH];
199     char *cmd;
200
201     if ((cmd = strrchr(cmdname, '/'))
202         if (cmd = strrchr(cmdname, '/'))
203             ++cmd;
204     else
205         cmd = cmdname;
206
207     /* get user type based on the program name */
208     if (strcmp(cmd, CMD_PREFIX_USER,
209             strlen(CMD_PREFIX_USER)) == 0)
210         strcpy(usertype, USERATTR_TYPE_NORMAL_KW);
211     else
212         strcpy(usertype, USERATTR_TYPE_NONADMIN_KW);
213
214     return (usertype);
215 }
216
217 _____unchanged_portion_omitted_____

```



```

*****
17396 Mon Sep  9 17:55:51 2013
new/usr/src/cmd/oamuser/user/useradd.c
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22  * Copyright 2008 Sun Microsystems, Inc. All rights reserved.
23  * Use is subject to license terms.
24 */

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

29 /*
30  * Copyright (c) 2013 RackTop Systems.
31  */
32 #endif /* ! codereview */

34 #include      <sys/types.h>
35 #include      <sys/stat.h>
36 #include      <sys/param.h>
37 #include      <stdio.h>
38 #include      <stdlib.h>
39 #include      <ctype.h>
40 #include      <limits.h>
41 #include      <string.h>
42 #include      <userdefs.h>
43 #include      <errno.h>
44 #include      <project.h>
45 #include      <unistd.h>
46 #include      <user_attr.h>
47 #include      <libcmdutils.h>
48 #endif /* ! codereview */
49 #include      "users.h"
50 #include      "messages.h"
51 #include      "userdisp.h"
52 #include      "funcs.h"

54 /*
55  * useradd [-u uid [-o] | -g group | -G group [[, group]...] | -d dir [-m]
56  *          | -s shell | -c comment | -k skel_dir | -b base_dir ]
57  *          [-A authorization [, authorization ...]]

```

```

58  *          [-P profile [, profile ...]]
59  *          [-K key=value ]
60  *          [-R role [, role ...]] [-p project [, project ...]] login
61  * useradd -D [ -g group ] [ -b base_dir | -f inactive | -e expire |
62  *          -s shell | -k skel_dir ]
63  *          [-A authorization [, authorization ...]]
64  *          [-P profile [, profile ...]] [-K key=value ]
65  *          [-R role [, role ...]] [-p project [, project ...]] login
66  *
67  *          This command adds new user logins to the system. Arguments are:
68  *
69  *          uid - an integer
70  *          group - an existing group's integer ID or char string name
71  *          dir - home directory
72  *          shell - a program to be used as a shell
73  *          comment - any text string
74  *          skel_dir - a skeleton directory
75  *          base_dir - a directory
76  *          login - a string of printable chars except colon(:)
77  *          authorization - One or more comma separated authorizations defined
78  *                        in auth_attr(4).
79  *          profile - One or more comma separated execution profiles defined
80  *                  in prof_attr(4)
81  *          role - One or more comma-separated role names defined in user_attr(4)
82  *          project - One or more comma-separated project names or numbers
83  *
84  */

86 extern struct userdefs *getusrdef();
87 extern void dispusrdef();

89 static void cleanup();

29 extern uid_t findnextuid(void);
91 extern int check_perm(), valid_expire();
92 extern int putusrdef(), valid_uid();
93 extern int call_passmgmt(), edit_group(), create_home();
94 extern int edit_project();
95 extern int **valid_lgroup();
96 extern projid_t **valid_lproject();
97 extern void update_def(struct userdefs *);
98 extern void import_def(struct userdefs *);

100 static uid_t uid; /* new uid */
101 static char *logname; /* login name to add */
102 static struct userdefs *usrdefs; /* defaults for useradd */

104 char *cmdname;

106 static char homedir[ PATH_MAX + 1 ]; /* home directory */
107 static char gidstring[32]; /* group id string representation */
108 static gid_t gid; /* gid of new login */
109 static char uidstring[32]; /* user id string representation */
110 static char *uidstr = NULL; /* uid from command line */
111 static char *base_dir = NULL; /* base_dir from command line */
112 static char *group = NULL; /* group from command line */
113 static char *grps = NULL; /* multi groups from command line */
114 static char *dir = NULL; /* home dir from command line */
115 static char *shell = NULL; /* shell from command line */
116 static char *comment = NULL; /* comment from command line */
117 static char *skel_dir = NULL; /* skel dir from command line */
118 static long inact; /* inactive days */
119 static char *inactstr = NULL; /* inactive from command line */
120 static char inactstring[10]; /* inactivity string representation */
121 static char *expirestr = NULL; /* expiration date from command line */
122 static char *projects = NULL; /* project id's from command line */

```

```

124 static char *usertype = NULL; /* type of user, either role or normal */

126 typedef enum {
127     BASEDIR = 0,
128     SKELDIR,
129     SHELL
130 } path_opt_t;

133 static void valid_input(path_opt_t, const char *);

135 int
136 main(argc, argv)
137 int argc;
138 char *argv[];
139 {
140     int ch, ret, mflag = 0, oflag = 0, Dflag = 0, **gidlist = NULL;
141     projid_t **projlist = NULL;
142     int ch, ret, mflag = 0, oflag = 0, Dflag = 0, **gidlist;
143     projid_t **projlist;
144     char *ptr; /* loc in a str, may be set by strtol */
145     struct group *g_ptr;
146     struct project p_ptr;
147     char mybuf[PROJECT_BUFSZ];
148     struct stat statbuf; /* status buffer for stat */
149     int warning;
150     int busy = 0;
151     char **nargv; /* arguments for execvp of passmgmt */
152     int argindex; /* argument index into nargv */

153     cmdname = argv[0];

154     if (geteuid() != 0) {
155         errmsg(M_PERM_DENIED);
156         exit(EX_NO_PERM);
157     }

158     opterr = 0; /* no print errors from getopt */
159     usertype = getusertype(argv[0]);

160     change_key(USERATTR_TYPE_KW, usertype);

161     while ((ch = getopt(argc, argv,
162         "b:c:Dd:e:f:G:g:k:mop:s:u:A:P:R:K:")) != EOF)
163     switch (ch) {
164     case 'b':
165         base_dir = optarg;
166         break;

167     case 'c':
168         comment = optarg;
169         break;

170     case 'D':
171         Dflag++;
172         break;

173     case 'd':
174         dir = optarg;
175         break;

176     case 'e':
177         expirestr = optarg;
178         break;

```

```

187     case 'f':
188         inactstr = optarg;
189         break;

190     case 'G':
191         grps = optarg;
192         break;

193     case 'g':
194         group = optarg;
195         break;

196     case 'k':
197         skel_dir = optarg;
198         break;

199     case 'm':
200         mflag++;
201         break;

202     case 'o':
203         oflag++;
204         break;

205     case 'p':
206         projects = optarg;
207         break;

208     case 's':
209         shell = optarg;
210         break;

211     case 'u':
212         uidstr = optarg;
213         break;

214     case 'A':
215         change_key(USERATTR_AUTHS_KW, optarg);
216         break;

217     case 'P':
218         change_key(USERATTR_PROFILES_KW, optarg);
219         break;

220     case 'R':
221         if (is_role(usertype)) {
222             errmsg(M_ARUSAGE);
223             exit(EX_SYNTAX);
224         }
225         change_key(USERATTR_ROLES_KW, optarg);
226         break;

227     case 'K':
228         change_key(NULL, optarg);
229         break;

230     default:
231     case '?':
232         if (is_role(usertype))
233             errmsg(M_ARUSAGE);
234         else
235             errmsg(M_AUSAGE);
236         exit(EX_SYNTAX);
237     }

238     /* get defaults for adding new users */

```

```

253     usrdefs = getusrdef(usertype);
254
255     if (Dflag) {
256         /* DISPLAY mode */
257
258         /* check syntax */
259         if (optind != argc) {
260             if (is_role(usertype))
261                 errmsg(M_ARUSAGE);
262             else
263                 errmsg(M_AUSAGE);
264             exit(EX_SYNTAX);
265         }
266
267         if (uidstr != NULL || oflag || grps != NULL ||
268             dir != NULL || mflag || comment != NULL) {
269             if (is_role(usertype))
270                 errmsg(M_ARUSAGE);
271             else
272                 errmsg(M_AUSAGE);
273             exit(EX_SYNTAX);
274         }
275
276         /* Group must be an existing group */
277         if (group != NULL) {
278             switch (valid_group(group, &g_ptr, &warning)) {
279             case INVALID:
280                 errmsg(M_INVALID, group, "group id");
281                 exit(EX_BADARG);
282                 /*NOTREACHED*/
283             case TOOBIG:
284                 errmsg(M_TOOBIG, "gid", group);
285                 exit(EX_BADARG);
286                 /*NOTREACHED*/
287             case RESERVED:
288             case UNIQUE:
289                 errmsg(M_GRP_NOTUSED, group);
290                 exit(EX_NAME_NOT_EXIST);
291             }
292             if (warning)
293                 warningmsg(warning, group);
294
295             usrdefs->defgroup = g_ptr->gr_gid;
296             usrdefs->defgname = g_ptr->gr_name;
297
298         }
299
300         /* project must be an existing project */
301         if (projects != NULL) {
302             switch (valid_project(projects, &p_ptr, mybuf,
303                 sizeof(mybuf), &warning)) {
304             case INVALID:
305                 errmsg(M_INVALID, projects, "project id");
306                 exit(EX_BADARG);
307                 /*NOTREACHED*/
308             case TOOBIG:
309                 errmsg(M_TOOBIG, "projid", projects);
310                 exit(EX_BADARG);
311                 /*NOTREACHED*/
312             case UNIQUE:
313                 errmsg(M_PROJ_NOTUSED, projects);
314                 exit(EX_NAME_NOT_EXIST);
315             }
316             if (warning)
317                 warningmsg(warning, projects);

```

```

319         usrdefs->defproj = p_ptr.pj_projid;
320         usrdefs->defprojname = p_ptr.pj_name;
321     }
322
323     /* base_dir must be an existing directory */
324     if (base_dir != NULL) {
325         valid_input(BASEDIR, base_dir);
326         usrdefs->defparent = base_dir;
327     }
328
329     /* inactivity period is an integer */
330     if (inactstr != NULL) {
331         /* convert inactstr to integer */
332         inact = strtol(inactstr, &ptr, 10);
333         if (*ptr || inact < 0) {
334             errmsg(M_INVALID, inactstr,
335                 "inactivity period");
336             exit(EX_BADARG);
337         }
338
339         usrdefs->definact = inact;
340     }
341
342     /* expiration string is a date, newer than today */
343     if (expirestr != NULL) {
344         if (*expirestr) {
345             if (valid_expire(expirestr, (time_t *)0)
346                 == INVALID) {
347                 errmsg(M_INVALID, expirestr,
348                     "expiration date");
349                 exit(EX_BADARG);
350             }
351             usrdefs->defexpire = expirestr;
352         } else
353             /* Unset the expiration date */
354             usrdefs->defexpire = "";
355     }
356
357     if (shell != NULL) {
358         valid_input(SHELL, shell);
359         usrdefs->defshell = shell;
360     }
361
362     if (skel_dir != NULL) {
363         valid_input(SKELDIR, skel_dir);
364         usrdefs->defskel = skel_dir;
365     }
366     update_def(usrdefs);
367
368     /* change defaults for useradd */
369     if (putusrdef(usrdefs, usertype) < 0) {
370         errmsg(M_UPDATE, "created");
371         exit(EX_UPDATE);
372     }
373
374     /* Now, display */
375     dispusrdef(stdout, (D_ALL & ~D_RID), usertype);
376     exit(EX_SUCCESS);
377 }
378
379 /* ADD mode */
380
381 /* check syntax */
382 if (optind != argc - 1 || (skel_dir != NULL && !mflag)) {
383     if (is_role(usertype))
384         errmsg(M_ARUSAGE);

```

```

385         else
386             errmsg(M_AUSAGE);
387         exit(EX_SYNTAX);
388     }
390     logname = argv[optind];
391     switch (valid_login(logname, (struct passwd **)NULL, &warning)) {
392     case INVALID:
393         errmsg(M_INVALID, logname, "login name");
394         exit(EX_BADARG);
395         /*NOTREACHED*/
397     case NOTUNIQUE:
398         errmsg(M_USED, logname);
399         exit(EX_NAME_EXISTS);
400         /*NOTREACHED*/
401     }
403     if (warning)
404         warningmsg(warning, logname);
405     if (uidstr != NULL) {
406         /* convert uidstr to integer */
407         errno = 0;
408         uid = (uid_t)strtol(uidstr, &ptr, (int)10);
409         if (*ptr || errno == ERANGE) {
410             errmsg(M_INVALID, uidstr, "user id");
411             exit(EX_BADARG);
412         }
414         switch (valid_uid(uid, NULL)) {
415         case NOTUNIQUE:
416             if (!oflag) {
417                 /* override not specified */
418                 errmsg(M_UID_USED, uid);
419                 exit(EX_ID_EXISTS);
420             }
421             break;
422         case RESERVED:
423             errmsg(M_RESERVED, uid);
424             break;
425         case TOOBIG:
426             errmsg(M_TOOBIG, "uid", uid);
427             exit(EX_BADARG);
428             break;
429         }
431     } else {
433         if (findnextuid(DEFRID+1, MAXUID, &uid) != 0) {
434             if ((uid = findnextuid()) < 0) {
435                 errmsg(M_INVALID, "default id", "user id");
436                 exit(EX_ID_EXISTS);
437             }
439         }
440         if (group != NULL) {
441             switch (valid_group(group, &g_ptr, &warning)) {
442             case INVALID:
443                 errmsg(M_INVALID, group, "group id");
444                 exit(EX_BADARG);
445                 /*NOTREACHED*/
446             case TOOBIG:
447                 errmsg(M_TOOBIG, "gid", group);
448                 exit(EX_BADARG);
449                 /*NOTREACHED*/
450             case RESERVED:

```

```

450         case UNIQUE:
451             errmsg(M_GRP_NOTUSED, group);
452             exit(EX_NAME_NOT_EXIST);
453             /*NOTREACHED*/
454         }
456         if (warning)
457             warningmsg(warning, group);
458         gid = g_ptr->gr_gid;
460     } else gid = usrdefs->defgroup;
462     if (grps != NULL) {
463         if (!*grps)
464             /* ignore -G "" */
465             grps = (char *)0;
466         else if (!(gidlist = valid_lgroup(grps, gid)))
467             exit(EX_BADARG);
468     }
470     if (projects != NULL) {
471         if (!*projects)
472             projects = (char *)0;
473         else if (!(projlist = valid_lproject(projects)))
474             exit(EX_BADARG);
475     }
477     /* if base_dir is provided, check its validity; otherwise default */
478     if (base_dir != NULL)
479         valid_input(BASEDIR, base_dir);
480     else
481         base_dir = usrdefs->defparent;
483     if (dir == NULL) {
484         /* set homedir to home directory made from base_dir */
485         (void) sprintf(homedir, "%s/%s", base_dir, logname);
487     } else if (REL_PATH(dir)) {
488         errmsg(M_RELPATH, dir);
489         exit(EX_BADARG);
491     } else
492         (void) strcpy(homedir, dir);
494     if (mflag) {
495         /* Does home dir. already exist? */
496         if (stat(homedir, &statbuf) == 0) {
497             /* directory exists - don't try to create */
498             mflag = 0;
500         }
501         if (check_perm(statbuf, uid, gid, S_IXOTH) != 0)
502             errmsg(M_NO_PERM, logname, homedir);
503     }
504     /*
505     * if shell, skel_dir are provided, check their validity.
506     * Otherwise default.
507     */
508     if (shell != NULL)
509         valid_input(SHELL, shell);
510     else
511         shell = usrdefs->defshell;
513     if (skel_dir != NULL)
514         valid_input(SKELDIR, skel_dir);
515     else

```

```

516         skel_dir = usrdefs->defskel;

518     if (inactstr != NULL) {
519         /* convert inactstr to integer */
520         inact = strtol(inactstr, &ptr, 10);
521         if (*ptr || inact < 0) {
522             errmsg(M_INVALID, inactstr, "inactivity period");
523             exit(EX_BADARG);
524         }
525     } else inact = usrdefs->definact;

527     /* expiration string is a date, newer than today */
528     if (expirestr != NULL) {
529         if (*expirestr) {
530             if (valid_expire(expirestr, (time_t *)0) == INVALID) {
531                 errmsg(M_INVALID, expirestr, "expiration date");
532                 exit(EX_BADARG);
533             }
534             usrdefs->defexpire = expirestr;
535         } else
536             /* Unset the expiration date */
537             expirestr = (char *)0;

539     } else expirestr = usrdefs->defexpire;

541     import_def(usrdefs);

543     /* must now call passmgmt */

545     /* set up arguments to passmgmt in nargv array */
546     nargv = malloc((30 + nkeys * 2) * sizeof(char *));
547     argindex = 0;
548     nargv[argindex++] = PASSMGMT;
549     nargv[argindex++] = "-a";      /* add */

551     if (comment != NULL) {
552         /* comment */
553         nargv[argindex++] = "-c";
554         nargv[argindex++] = comment;
555     }

557     /* flags for home directory */
558     nargv[argindex++] = "-h";
559     nargv[argindex++] = homedir;

561     /* set gid flag */
562     nargv[argindex++] = "-g";
563     (void) sprintf(gidstring, "%u", gid);
564     nargv[argindex++] = gidstring;

566     /* shell */
567     nargv[argindex++] = "-s";
568     nargv[argindex++] = shell;

570     /* set inactive */
571     nargv[argindex++] = "-f";
572     (void) sprintf(inactstring, "%ld", inact);
573     nargv[argindex++] = inactstring;

575     /* set expiration date */
576     if (expirestr != NULL) {
577         nargv[argindex++] = "-e";
578         nargv[argindex++] = expirestr;
579     }

581     /* set uid flag */

```

```

582     nargv[argindex++] = "-u";
583     (void) sprintf(uidstring, "%u", uid);
584     nargv[argindex++] = uidstring;

586     if (oflag) nargv[argindex++] = "-o";

588     if (nkeys > 1)
589         addkey_args(nargv, &argindex);

591     /* finally - login name */
592     nargv[argindex++] = logname;

594     /* set the last to null */
595     nargv[argindex++] = NULL;

597     /* now call passmgmt */
598     ret = PEX_FAILED;
599     /*
600     * If call_passmgmt fails for any reason other than PEX_BADUID, exit
601     * is invoked with an appropriate error message. If PEX_BADUID is
602     * returned, then if the user specified the ID, exit is invoked
603     * with an appropriate error message. Otherwise we try to pick a
604     * different ID and try again. If we run out of IDs, i.e. no more
605     * users can be created, then -1 is returned and we terminate via exit.
606     * If PEX_BUSY is returned we increment a count, since we will stop
607     * trying if PEX_BUSY reaches 3. For PEX_SUCCESS we immediately
608     * terminate the loop.
609     */
610     while (busy < 3 && ret != PEX_SUCCESS) {
611         switch (ret = call_passmgmt(nargv)) {
612             case PEX_SUCCESS:
613                 break;
614             case PEX_BUSY:
615                 busy++;
616                 break;
617             case PEX_HOSED_FILES:
618                 errmsg(M_HOSED_FILES);
619                 exit(EX_INCONSISTENT);
620                 break;

622             case PEX_SYNTAX:
623             case PEX_BADARG:
624                 /* should NEVER occur that passmgmt usage is wrong */
625                 if (is_role(usertype))
626                     errmsg(M_ARUSAGE);
627                 else
628                     errmsg(M_AUSAGE);
629                 exit(EX_SYNTAX);
630                 break;

632             case PEX_BADUID:
633                 /*
634                 * The uid has been taken. If it was specified by a
635                 * user, then we must fail. Otherwise, keep trying
636                 * to get a good uid until we run out of IDs.
637                 */
638                 if (uidstr != NULL) {
639                     errmsg(M_UID_USED, uid);
640                     exit(EX_ID_EXISTS);
641                 } else {
642                     if (findnextuid(DEFRID+1, MAXUID, &uid) != 0) {
643                         if ((uid = findnextuid()) < 0) {
644                             errmsg(M_INVALID, "default id",
645                                 "user id");
646                             exit(EX_ID_EXISTS);
647                         }
648                     }
649                 }
650             }

```

```
647         (void) sprintf(uidstring, "%u", uid);
648     }
649     break;

651     case PEX_BADNAME:
652         /* invalid loname */
653         errmsg(M_USED, logname);
654         exit(EX_NAME_EXISTS);
655     break;

657     default:
658         errmsg(M_UPDATE, "created");
659         exit(ret);
660     break;
661 }
662 }
663 if (busy == 3) {
664     errmsg(M_UPDATE, "created");
665     exit(ret);
666 }

668 /* add group entry */
669 if ((grps != NULL) && edit_group(logname, (char *)0, gidlist, 0)) {
670     errmsg(M_UPDATE, "created");
671     cleanup(logname);
672     exit(EX_UPDATE);
673 }

675 /* update project database */
676 if ((projects != NULL) &&
677     edit_project(logname, (char *)NULL, projlist, 0)) {
678     errmsg(M_UPDATE, "created");
679     cleanup(logname);
680     exit(EX_UPDATE);
681 }

683 /* create home directory */
684 if (mflag &&
685     (create_home(homedir, skel_dir, uid, gid) != EX_SUCCESS)) {
686     (void) edit_group(logname, (char *)0, (int **)0, 1);
687     cleanup(logname);
688     exit(EX_HOMEDIR);
689 }

691     return (ret);
692 }
unchanged_portion_omitted
```

new/usr/src/cmd/oamuser/user/userdefs.c

1

```
*****
9936 Mon Sep  9 17:55:52 2013
new/usr/src/cmd/oamuser/user/userdefs.c
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License, Version 1.0 only
6  * (the "License"). You may not use this file except in compliance
7  * with the License.
8  *
9  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 * or http://www.opensolaris.org/os/licensing.
11 * See the License for the specific language governing permissions
12 * and limitations under the License.
13 *
14 * When distributing Covered Code, include this CDDL HEADER in each
15 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 * If applicable, add the following below this CDDL HEADER, with the
17 * fields enclosed by brackets "[]" replaced with your own identifying
18 * information: Portions Copyright [yyyy] [name of copyright owner]
19 *
20 * CDDL HEADER END
21 */

23 /*
24  * Copyright 2005 Sun Microsystems, Inc. All rights reserved.
25  * Use is subject to license terms.
26 */

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

31 /*
32  * Copyright (c) 2013 RackTop Systems.
33  */

32 #pragma ident      "%Z%M% %I%      %E% SMI"

35 /*LINTLIBRARY*/

37 #include      <sys/types.h>
38 #include      <stdio.h>
39 #include      <string.h>
40 #include      <userdefs.h>
41 #include      <user_attr.h>
42 #include      <limits.h>
43 #include      <stdlib.h>
44 #include      <stddef.h>
45 #include      <time.h>
46 #include      <unistd.h>
47 #include      "userdisp.h"
48 #include      "funcs.h"
49 #include      "messages.h"

51 /* Print out a NL when the line gets too long */
52 #define PRINTNL() \
53     if (outcount > 40) { \
54         outcount = 0; \
55         (void) fprintf(fp, "\n"); \
```

new/usr/src/cmd/oamuser/user/userdefs.c

2

```
56     }

58 #define SKIPWS(ptr)      while (*ptr && (*ptr == ' ' || *ptr == '\t')) ptr++
57 #define SKIPWS(ptr)      while (*ptr && *ptr == ' ' || *ptr == '\t') ptr++

60 static char *dup_to_nl(char *);

62 static struct userdefs defaults = {
63     DEFRID, DEFGROUP, DEFGNAME, DEFPARENT, DEFSKEL,
64     DEFSHL, DEFINACT, DEFEXPIRE, DEFAUTH, DEFPROF,
65     DEFROLE, DEFPROJ, DEFPROJNAME, DEFIMPRIV,
66     DEFDFLTPRIV, DEFLOCK_AFTER_RETRIES
67 };
unchanged_portion_omitted
```

new/usr/src/cmd/oamuser/user/usermod.c

1

```
*****
15592 Mon Sep  9 17:55:52 2013
new/usr/src/cmd/oamuser/user/usermod.c
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22  * Copyright 2008 Sun Microsystems, Inc. All rights reserved.
23  * Use is subject to license terms.
24 */
25
26 /*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
27 /*      All Rights Reserved      */
28
29 /*
30  * Copyright (c) 2013 RackTop Systems.
31  */
32
33 #include <sys/types.h>
34 #include <sys/stat.h>
35 #include <sys/param.h>
36 #include <stdio.h>
37 #include <stdlib.h>
38 #include <ctype.h>
39 #include <limits.h>
40 #include <string.h>
41 #include <userdefs.h>
42 #include <user_attr.h>
43 #include <nss_dbdefs.h>
44 #include <errno.h>
45 #include <project.h>
46 #include "users.h"
47 #include "messages.h"
48 #include "funcs.h"
49
50 /*
51  * usermod [-u uid [-o | -g group | -G group [[,group]...]] | -d dir [-m
52  *          | -s shell | -c comment | -l new_logname]
53  *          | -f inactive | -e expire ]
54  *          [ -A authorization [, authorization ...]]
55  *          [ -P profile [, profile ...]]
56  *          [ -R role [, role ...]]
```

new/usr/src/cmd/oamuser/user/usermod.c

2

```
57 *          [ -K key=value ]
58 *          [ -p project [, project]] login
59 *
60 * This command adds new user logins to the system. Arguments are:
61 *
62 * uid - an integer less than MAXUID
63 * group - an existing group's integer ID or char string name
64 * dir - a directory
65 * shell - a program to be used as a shell
66 * comment - any text string
67 * skel_dir - a directory
68 * base_dir - a directory
69 * rid - an integer less than 2**16 (USHORT)
70 * login - a string of printable chars except colon (:)
71 * inactive - number of days a login maybe inactive before it is locked
72 * expire - date when a login is no longer valid
73 * authorization - One or more comma separated authorizations defined
74 *                 in auth_attr(4).
75 * profile - One or more comma separated execution profiles defined
76 *           in prof_attr(4)
77 * role - One or more comma-separated role names defined in user_attr(4)
78 * key=value - One or more -K options each specifying a valid user_attr(4)
79 *             attribute.
80 *
81 */
82
83 extern int **valid_lgroup(), isbusy();
84 extern int valid_uid(), check_perm(), create_home(), move_dir();
85 extern int valid_expire(), edit_group(), call_passmgmt();
86 extern projid_t **valid_lproject();
87
88 static uid_t uid; /* new uid */
89 static gid_t gid; /* gid of new login */
90 static char *new_logname = NULL; /* new login name with -l option */
91 static char *uidstr = NULL; /* uid from command line */
92 static char *group = NULL; /* group from command line */
93 static char *grps = NULL; /* multi groups from command line */
94 static char *dir = NULL; /* home dir from command line */
95 static char *shell = NULL; /* shell from command line */
96 static char *comment = NULL; /* comment from command line */
97 static char *logname = NULL; /* login name to add */
98 static char *inactstr = NULL; /* inactive from command line */
99 static char *expire = NULL; /* expiration date from command line */
100 static char *projects = NULL; /* project ids from command line */
101 static char *usertype;
102
103 char *cmdname;
104 static char gidstring[32], uidstring[32];
105 char inactstring[10];
106
107 char *
108 strcpmalloc(str)
109 char *str;
110 {
111     if (str == NULL)
112         return (NULL);
113
114     return (strdup(str));
115 }
116
117 unchanged_portion_omitted
118
119 int
120 main(argc, argv)
121 int argc;
122 char **argv;
123 {
```



```

146     int ch, ret = EX_SUCCESS, call_pass = 0, oflag = 0;
147     int tries, mflag = 0, inact, **gidlist, flag = 0;
148     boolean_t fail_if_busy = B_FALSE;
149     char *ptr;
150     struct passwd *pstruct;      /* password struct for login */
151     struct passwd *pw;
152     struct group *g_ptr;        /* validated group from -g */
153     struct stat statbuf;        /* status buffer for stat */
154 #ifndef att
155     FILE *pwf;                  /* fille ptr for opened passwd file */
156 #endif
157     int warning;
158     projid_t **projlist;
159     char **nargv;                /* arguments for execvp of passgmt */
160     int argindex;                /* argument index into nargv */
161     userattr_t *ua;
162     char *val;
163     int isrole;                  /* current account is role */

164
165     cmdname = argv[0];

166
167     if (geteuid() != 0) {
168         errmsg(M_PERM_DENIED);
169         exit(EX_NO_PERM);
170     }

171
172     opterr = 0;                  /* no print errors from getopt */
173     /* get user type based on the program name */
174     usertype = getusertype(argv[0]);

175
176     while ((ch = getopt(argc, argv,
177         "c:d:e:f:G:g:l:mop:s:u:A:P:R:K:") != EOF)
178     {
179         switch (ch) {
180             case 'c':
181                 comment = optarg;
182                 flag++;
183                 break;
184             case 'd':
185                 dir = optarg;
186                 fail_if_busy = B_TRUE;
187                 flag++;
188                 break;
189             case 'e':
190                 expire = optarg;
191                 flag++;
192                 break;
193             case 'f':
194                 inactstr = optarg;
195                 flag++;
196                 break;
197             case 'G':
198                 grps = optarg;
199                 flag++;
200                 break;
201             case 'g':
202                 group = optarg;
203                 fail_if_busy = B_TRUE;
204                 flag++;
205                 break;
206             case 'l':
207                 new_logname = optarg;
208                 fail_if_busy = B_TRUE;
209                 flag++;
210                 break;
211             case 'm':
212                 mflag++;

```

```

212         flag++;
213         fail_if_busy = B_TRUE;
214         break;
215     case 'o':
216         oflag++;
217         flag++;
218         fail_if_busy = B_TRUE;
219         break;
220     case 'p':
221         projects = optarg;
222         flag++;
223         break;
224     case 's':
225         shell = optarg;
226         flag++;
227         break;
228     case 'u':
229         uidstr = optarg;
230         flag++;
231         fail_if_busy = B_TRUE;
232         break;
233     case 'A':
234         change_key(USERATTR_AUTHS_KW, optarg);
235         flag++;
236         break;
237     case 'P':
238         change_key(USERATTR_PROFILES_KW, optarg);
239         flag++;
240         break;
241     case 'R':
242         change_key(USERATTR_ROLES_KW, optarg);
243         flag++;
244         break;
245     case 'K':
246         change_key(NULL, optarg);
247         flag++;
248         break;
249     default:
250     case '?':
251         if (is_role(usertype))
252             errmsg(M_MRUSAGE);
253         else
254             errmsg(M_MUSAGE);
255         exit(EX_SYNTAX);
256     }

257
258     if (optind != argc - 1 || flag == 0) {
259         if (is_role(usertype))
260             errmsg(M_MRUSAGE);
261         else
262             errmsg(M_MUSAGE);
263         exit(EX_SYNTAX);
264     }

265
266     if ((!uidstr && oflag) || (mflag && !dir)) {
267         if (is_role(usertype))
268             errmsg(M_MRUSAGE);
269         else
270             errmsg(M_MUSAGE);
271         exit(EX_SYNTAX);
272     }

273
274     logname = argv[optind];

275
276     /* Determine whether the account is a role or not */
277     if ((ua = getusernam(logname)) == NULL ||

```

```

278     (val = kva_match(ua->attr, USERATTR_TYPE_KW)) == NULL ||
279     strcmp(val, USERATTR_TYPE_NONADMIN_KW) != 0)
280     isrole = 0;
281 else
282     isrole = 1;

284 /* Verify that rolemod is used for roles and usermod for users */
285 if (isrole != is_role(usertype)) {
286     if (isrole)
287         errmsg(M_ISROLE);
288     else
289         errmsg(M_ISUSER);
290     exit(EX_SYNTAX);
291 }

293 /* Set the usertype key; defaults to the commandline */
294 usertype = getsetdefval(USERATTR_TYPE_KW, usertype);

296 if (is_role(usertype)) {
297     /* Roles can't have roles */
298     if (getsetdefval(USERATTR_ROLES_KW, NULL) != NULL) {
299         errmsg(M_MRUSAGE);
300         exit(EX_SYNTAX);
301     }
302     /* If it was an ordinary user, delete its roles */
303     if (!isrole)
304         change_key(USERATTR_ROLES_KW, "");
305 }

307 #ifdef att
308     pw = getpwnam(logname);
309 #else
310     /*
311     * Do this with fgetpwent to make sure we are only looking on local
312     * system (since passmgmt only works on local system).
313     */
314     if ((pwf = fopen("/etc/passwd", "r")) == NULL) {
315         errmsg(M_OOPS, "open", "/etc/passwd");
316         exit(EX_FAILURE);
317     }
318     while ((pw = fgetpwent(pwf)) != NULL)
319         if (strcmp(pw->pw_name, logname) == 0)
320             break;

322     fclose(pwf);
323 #endif

325 if (pw == NULL) {
326     char        pwdb[NSS_BUFLen_PASSWD];
327     struct passwd    pwd;

329     if (getpwnam_r(logname, &pwd, pwdb, sizeof (pwdb)) == NULL) {
330         /* This user does not exist. */
331         errmsg(M_EXIST, logname);
332         exit(EX_NAME_NOT_EXIST);
333     } else {
334         /* This user exists in non-local name service. */
335         errmsg(M_NONLOCAL, logname);
336         exit(EX_NOT_LOCAL);
337     }
338 }

340 pstruct = passwd_cpmmalloc(pw);

342 /*
343  * We can't modify a logged in user if any of the following

```

```

344     * are being changed:
345     * uid (-u & -o), group (-g), home dir (-m), loginname (-l).
346     * If none of those are specified it is okay to go ahead
347     * some types of changes only take effect on next login, some
348     * like authorisations and profiles take effect instantly.
349     * One might think that -K type=role should require that the
350     * user not be logged in, however this would make it very
351     * difficult to make the root account a role using this command.
352     */
353 if (isbusy(logname)) {
354     if (fail_if_busy) {
355         errmsg(M_BUSY, logname, "change");
356         exit(EX_BUSY);
357     }
358     warnmsg(WARN_LOGGED_IN, logname);
359 }

361 if (new_logname && strcmp(new_logname, logname)) {
362     switch (valid_login(new_logname, (struct passwd *)NULL,
363         &warning)) {
364     case INVALID:
365         errmsg(M_INVALID, new_logname, "login name");
366         exit(EX_BADARG);
367         /*NOTREACHED*/

369     case NOTUNIQUE:
370         errmsg(M_USED, new_logname);
371         exit(EX_NAME_EXISTS);
372         /*NOTREACHED*/

373     default:
374         call_pass = 1;
375         break;
376     }
377     if (warning)
378         warnmsg(warning, logname);
379 }

381 if (uidstr) {
382     /* convert uidstr to integer */
383     errno = 0;
384     uid = (uid_t)strtol(uidstr, &ptr, (int)10);
385     if (*ptr || errno == ERANGE) {
386         errmsg(M_INVALID, uidstr, "user id");
387         exit(EX_BADARG);
388     }

390     if (uid != pstruct->pw_uid) {
391         switch (valid_uid(uid, NULL)) {
392         case NOTUNIQUE:
393             if (!oflag) {
394                 /* override not specified */
395                 errmsg(M_UID_USED, uid);
396                 exit(EX_ID_EXISTS);
397             }
398             break;
399         case RESERVED:
400             errmsg(M_RESERVED, uid);
401             break;
402         case TOOBIG:
403             errmsg(M_TOOBIG, "uid", uid);
404             exit(EX_BADARG);
405             break;
406         }

408         call_pass = 1;

```

```

410     } else {
411         /* uid's the same, so don't change anything */
412         uidstr = NULL;
413         oflag = 0;
414     }
415
416 } else uid = pstruct->pw_uid;
417
418 if (group) {
419     switch (valid_group(group, &g_ptr, &warning)) {
420     case INVALID:
421         errmsg(M_INVALID, group, "group id");
422         exit(EX_BADARG);
423         /*NOTREACHED*/
424     case TOOBIG:
425         errmsg(M_TOOBIG, "gid", group);
426         exit(EX_BADARG);
427         /*NOTREACHED*/
428     case UNIQUE:
429         errmsg(M_GRP_NOTUSED, group);
430         exit(EX_NAME_NOT_EXIST);
431         /*NOTREACHED*/
432     case RESERVED:
433         gid = (gid_t)strtol(group, &ptr, (int)10);
434         errmsg(M_RESERVED_GID, gid);
435         break;
436     }
437     if (warning)
438         warningmsg(warning, group);
439
440     if (g_ptr != NULL)
441         gid = g_ptr->gr_gid;
442     else
443         gid = pstruct->pw_gid;
444
445     /* call passmgmt if gid is different, else ignore group */
446     if (gid != pstruct->pw_gid)
447         call_pass = 1;
448     else group = NULL;
449
450 } else gid = pstruct->pw_gid;
451
452 if (grps && *grps) {
453     if (!(gidlist = valid_lgroup(grps, gid)))
454         exit(EX_BADARG);
455 } else
456     gidlist = (int **)0;
457
458 if (projects && *projects) {
459     if (!(projlist = valid_lproject(projects)))
460         exit(EX_BADARG);
461 } else
462     projlist = (projid_t **)0;
463
464 if (dir) {
465     if (REL_PATH(dir)) {
466         errmsg(M_RELPATH, dir);
467         exit(EX_BADARG);
468     }
469     if (strcmp(pstruct->pw_dir, dir) == 0) {
470         /* home directory is the same so ignore dflag & mflag */
471         dir = NULL;
472         mflag = 0;
473     } else call_pass = 1;
474 }

```

```

476     if (mflag) {
477         if (stat(dir, &statbuf) == 0) {
478             /* Home directory exists */
479             if (check_perm(statbuf, pstruct->pw_uid,
480                 pstruct->pw_gid, S_IWOTH|S_IXOTH) != 0) {
481                 errmsg(M_NO_PERM, logname, dir);
482                 exit(EX_NO_PERM);
483             }
484         }
485     } else ret = create_home(dir, NULL, uid, gid);
486
487     if (ret == EX_SUCCESS)
488         ret = move_dir(pstruct->pw_dir, dir, logname);
489
490     if (ret != EX_SUCCESS)
491         exit(ret);
492 }
493
494 if (shell) {
495     if (REL_PATH(shell)) {
496         errmsg(M_RELPATH, shell);
497         exit(EX_BADARG);
498     }
499     if (strcmp(pstruct->pw_shell, shell) == 0) {
500         /* ignore s option if shell is not different */
501         shell = NULL;
502     } else {
503         if (stat(shell, &statbuf) < 0 ||
504             (statbuf.st_mode & S_IFMT) != S_IFREG ||
505             (statbuf.st_mode & 0555) != 0555) {
506             errmsg(M_INVALID, shell, "shell");
507             exit(EX_BADARG);
508         }
509     }
510
511     call_pass = 1;
512 }
513
514
515 if (comment) {
516     if (comment)
517         /* ignore comment if comment is not changed */
518         if (strcmp(pstruct->pw_comment, comment))
519             call_pass = 1;
520     else
521         comment = NULL;
522 #endif /* ! codereview */
523
524 /* inactive string is a positive integer */
525 if (inactstr) {
526     /* convert inactstr to integer */
527     inact = (int)strtol(inactstr, &ptr, 10);
528     if (*ptr || inact < 0) {
529         errmsg(M_INVALID, inactstr, "inactivity period");
530         exit(EX_BADARG);
531     }
532     call_pass = 1;
533 }
534
535 /* expiration string is a date, newer than today */
536 if (expire) {
537     if (*expire &&
538         valid_expire(expire, (time_t *)0) == INVALID) {
539         errmsg(M_INVALID, expire, "expiration date");
540         exit(EX_BADARG);

```

```

541     }
542     call_pass = 1;
543 }
544
545 if (nkeys > 0)
546     call_pass = 1;
547
548 /* that's it for validations - now do the work */
549
550 if (grps) {
551     /* redefine login's supplementary group memberships */
552     ret = edit_group(logname, new_logname, gidlist, 1);
553     if (ret != EX_SUCCESS) {
554         errmsg(M_UPDATE, "modified");
555         exit(ret);
556     }
557 }
558 if (projects) {
559     ret = edit_project(logname, (char *)NULL, projlist, 0);
560     if (ret != EX_SUCCESS) {
561         errmsg(M_UPDATE, "modified");
562         exit(ret);
563     }
564 }
565
566 if (!call_pass) exit(ret);
567
568 /* only get to here if need to call passmgmt */
569 /* set up arguments to passmgmt in nargv array */
570 nargv = malloc((30 + nkeys * 2) * sizeof(char *));
571
572 argindex = 0;
573 nargv[argindex++] = PASSMGMT;
574 nargv[argindex++] = "-m"; /* modify */
575
576 if (comment) { /* comment */
577     nargv[argindex++] = "-c";
578     nargv[argindex++] = comment;
579 }
580
581 if (dir) {
582     /* flags for home directory */
583     nargv[argindex++] = "-h";
584     nargv[argindex++] = dir;
585 }
586
587 if (group) {
588     /* set gid flag */
589     nargv[argindex++] = "-g";
590     (void) sprintf(gidstring, "%u", gid);
591     nargv[argindex++] = gidstring;
592 }
593
594 if (shell) { /* shell */
595     nargv[argindex++] = "-s";
596     nargv[argindex++] = shell;
597 }
598
599 if (inactstr) {
600     nargv[argindex++] = "-f";
601     nargv[argindex++] = inactstr;
602 }
603
604 if (expire) {
605     nargv[argindex++] = "-e";

```

```

607     nargv[argindex++] = expire;
608 }
609
610 if (uidstr) { /* set uid flag */
611     nargv[argindex++] = "-u";
612     (void) sprintf(uidstring, "%u", uid);
613     nargv[argindex++] = uidstring;
614 }
615
616 if (oflag) nargv[argindex++] = "-o";
617
618 if (new_logname) { /* redefine login name */
619     nargv[argindex++] = "-l";
620     nargv[argindex++] = new_logname;
621 }
622
623 if (nkeys > 0)
624     addkey_args(nargv, &argindex);
625
626 /* finally - login name */
627 nargv[argindex++] = logname;
628
629 /* set the last to null */
630 nargv[argindex++] = NULL;
631
632 /* now call passmgmt */
633 ret = PEX_FAILED;
634 for (tries = 3; ret != PEX_SUCCESS && tries--;) {
635     switch (ret = call_passmgmt(nargv)) {
636     case PEX_SUCCESS:
637     case PEX_BUSY:
638         break;
639
640     case PEX_HOSED_FILES:
641         errmsg(M_HOSED_FILES);
642         exit(EX_INCONSISTENT);
643         break;
644
645     case PEX_SYNTAX:
646     case PEX_BADARG:
647         /* should NEVER occur that passmgmt usage is wrong */
648         if (is_role(usertype))
649             errmsg(M_MRUSAGE);
650         else
651             errmsg(M_MUSAGE);
652         exit(EX_SYNTAX);
653         break;
654
655     case PEX_BADUID:
656         /* uid in use - shouldn't happen print message anyway */
657         errmsg(M_UID_USED, uid);
658         exit(EX_ID_EXISTS);
659         break;
660
661     case PEX_BADNAME:
662         /* invalid loname */
663         errmsg(M_USED, logname);
664         exit(EX_NAME_EXISTS);
665         break;
666
667     default:
668         errmsg(M_UPDATE, "modified");
669         exit(ret);
670         break;
671     }
672 }

```

new/usr/src/cmd/oamuser/user/usermod.c

11

```
673     if (tries == 0) {
674         errmsg(M_UPDATE, "modified");
675     }
677     exit(ret);
678     /*NOTREACHED*/
679 }
```

```

*****
2747 Mon Sep  9 17:55:53 2013
new/usr/src/cmd/oamuser/user/val_lgrp.c
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22  * Copyright 2009 Sun Microsystems, Inc. All rights reserved.
23  * Use is subject to license terms.
24 */

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

29 /*
30  * Copyright (c) 2013 RackTop Systems.
31  */

33 #endif /* ! codereview */

35 #include <sys/types.h>
36 #include <stdio.h>
37 #include <stdlib.h>
38 #include <sys/param.h>
39 #include <unistd.h>
40 #include <users.h>
41 #include <userdefs.h>
42 #include "messages.h"

44 extern void exit();
45 extern char *strtok();

47 static gid_t *grplist;
48 static int ngroups_max = 0;

50 /* Validate a list of groups */
51 int
52 valid_lgroup(char *list, gid_t gid)
53 {
54     int n_invalid = 0, i = 0, j;
55     char *ptr;
56     struct group *g_ptr;
57     int warning;

```

```

58     int dup_prim = 0; /* we don't duplicate our primary as a supplemental */
59
60     if( !list || !*list )
61         return( (int **) NULL );
62
63     if (ngroups_max == 0) {
64         ngroups_max = sysconf(_SC_NGROUPS_MAX);
65         grplist = malloc((ngroups_max + 1) * sizeof (gid_t));
66     }
67
68     while ((ptr = strtok((i || n_invalid || dup_prim)? NULL: list, ",")) {
69         while (ptr = strtok((i || n_invalid || dup_prim)? NULL: list, ",")) {
70
71             switch (valid_group(ptr, &g_ptr, &warning)) {
72             case INVALID:
73                 errmsg( M_INVALID, ptr, "group id" );
74                 n_invalid++;
75                 break;
76             case TOOBIG:
77                 errmsg( M_TOOBIG, "gid", ptr );
78                 n_invalid++;
79                 break;
80             case UNIQUE:
81                 errmsg( M_GRP_NOTUSED, ptr );
82                 n_invalid++;
83                 break;
84             case NOTUNIQUE:
85                 /* ignore duplicated primary */
86                 if (g_ptr->gr_gid == gid) {
87                     if (!dup_prim)
88                         dup_prim++;
89                     continue;
90                 }
91
92                 if( !i )
93                     grplist[ i++ ] = g_ptr->gr_gid;
94                 else {
95                     /* Keep out duplicates */
96                     for( j = 0; j < i; j++ )
97                         if( g_ptr->gr_gid == grplist[j] )
98                             break;
99
100                     if( j == i )
101                         /* Not a duplicate */
102                         grplist[i++] = g_ptr->gr_gid;
103                 }
104                 break;
105             }
106             if (warning)
107                 warningmsg(warning, ptr);
108
109             if( i >= ngroups_max ) {
110                 errmsg( M_MAXGROUPS, ngroups_max );
111                 break;
112             }
113         }
114     }
115
116     /* Terminate the list */
117     grplist[ i ] = -1;
118
119     if( n_invalid )
120         exit( EX_BADARG );
121
122     return( (int **)grplist );

```

new/usr/src/cmd/oamuser/user/val\_lprj.c

1

```
*****
2456 Mon Sep  9 17:55:54 2013
new/usr/src/cmd/oamuser/user/val_lprj.c
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
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 2005 Sun Microsystems, Inc. All rights reserved.
24 * Use is subject to license terms.
25 */
27 /*
28  * Copyright (c) 2013 RackTop Systems.
29  */
27 #pragma ident      "%Z%M% %I%      %E% SMI"

31 #include      <sys/types.h>
32 #include      <stdio.h>
33 #include      <strings.h>
34 #include      <sys/param.h>
35 #include      <users.h>
36 #include      <userdefs.h>
37 #include      <project.h>
38 #include      <errno.h>
39 #include      "messages.h"

42 static projid_t projlist[NPROJECTS_MAX + 1];
43 static int nproj_max = NPROJECTS_MAX;

45 /* Validate a list of projects */
46 int **
47 valid_lproject(char *list)
48 {
49     int n_invalid = 0;
50     int i = 0;
51     int j;
52     char *ptr;
53     struct project projent;
54     int warning;
55     char mybuf[PROJECT_BUFSZ];
```

new/usr/src/cmd/oamuser/user/val\_lprj.c

2

```
57     if (!list || !*list)
58         return ((int **)NULL);

60     while ((ptr = strtok((i || n_invalid) ? NULL : list, ",")) {
58     while (ptr = strtok((i || n_invalid) ? NULL : list), ",") {

62         switch (valid_project(ptr, &projent, mybuf, sizeof (mybuf),
63             &warning)) {
64             case INVALID:
65                 errmsg(M_INVALID, ptr, "project id");
66                 n_invalid++;
67                 break;
68             case TOOBIG:
69                 errmsg(M_TOOBIG, "projid", ptr);
70                 n_invalid++;
71                 break;
72             case UNIQUE:
73                 errmsg(M_PROJ_NOTUSED, ptr);
74                 n_invalid++;
75                 break;
76             case NOTUNIQUE:
77                 if (!i)
78                     /* ignore respecified primary */
79                     projlist[i++] = projent.pj_projid;
80                 else {
81                     /* Keep out duplicates */
82                     for (j = 0; j < i; j++)
83                         if (projent.pj_projid == projlist[j])
84                             break;

86                     if (j == i)
87                         /* Not a duplicate */
88                         projlist[i++] = projent.pj_projid;
89                 }
90                 break;
91             }
92             if (warning)
93                 warningmsg(warning, ptr);

95             if (i >= nproj_max) {
96                 errmsg(M_MAXPROJECTS, nproj_max);
97                 break;
98             }
99         }

101     /* Terminate the list */
102     projlist[i] = -1;

104     if (n_invalid)
105         exit(EX_BADARG);

107     return ((int **)projlist);
108 }
unchanged_portion_omitted
```

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

1

```
*****
1675 Mon Sep  9 17:55:55 2013
new/usr/src/lib/libcmdutils/Makefile.com
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 2003, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright (c) 2013 RackTop Systems.
24 #endif /* ! codereview */
25 #

27 LIBRARY=      libcmdutils.a
28 VERS=         .1
29 CMD_OBJS=     avltree.o sysattrs.o writefile.o process_xattrs.o uid.o gid.o
23 CMD_OBJS=     avltree.o sysattrs.o writefile.o process_xattrs.o
30 COM_OBJS=     list.o
31 OBJECTS=      $(CMD_OBJS) $(COM_OBJS)

33 include ../../Makefile.lib
34 include ../../Makefile.rootfs

36 LIBS =        $(DYNLIB) $(LINTLIB)

38 LDLIBS +=     -lc -lavl -lnvpair

40 SRCDIR =      ../common

42 COMDIR= $(SRC)/common/list
43 SRCS= \
44   $(CMD_OBJS:%.o=$(SRCDIR)/%.c) \
45   $(COM_OBJS:%.o=$(COMDIR)/%.c)

47 $(LINTLIB) := SRCS = $(SRCDIR)/$(LINTSRC)

49 CFLAGS +=     $(CCVERBOSE)

51 # All commands using the common avltree interfaces must
52 # be largefile aware.
53 CPPFLAGS +=   -I.. -I../common/inc -D_REENTRANT -D_FILE_OFFSET_BITS=64

55 .KEEP_STATE:
```

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

2

```
57 all: $(LIBS)

59 lint: lintcheck

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

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



```

*****
1788 Mon Sep  9 17:55:56 2013
new/usr/src/lib/libcndutils/common/mapfile-vers
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
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) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright (c) 2013 RackTop Systems.
24 #endif /* ! codereview */
25 #

27 #
28 # MAPFILE HEADER START
29 #
30 # WARNING: STOP NOW. DO NOT MODIFY THIS FILE.
31 # Object versioning must comply with the rules detailed in
32 #
33 #     usr/src/lib/README.mapfiles
34 #
35 # You should not be making modifications here until you've read the most current
36 # copy of that file. If you need help, contact a gatekeeper for guidance.
37 #
38 # MAPFILE HEADER END
39 #

41 $mapfile_version 2

43 SYMBOL_VERSION SUNWprivate_1.1 {
44     global:
45         add_tnode;
46         destroy_tree;
47         findnextgid;
48         findnextuid;
49 #endif /* ! codereview */
50     list_create;
51     list_destroy;
52     list_head;
53     list_insert_after;
54     list_insert_before;
55     list_insert_head;
56     list_insert_tail;
57     list_is_empty;

```

```

58     list_link_active;
59     list_link_init;
60     list_move_tail;
61     list_next;
62     list_prev;
63     list_remove;
64     list_remove_head;
65     list_remove_tail;
66     list_tail;
67     tnode_compare;
68     sysattr_type;
69     sysattr_support;
70     writefile;
71     get_attrdirs;
72     mv_xattrs;
73     sysattr_list;
74     local:
75         *;
76 };

```

```

*****
2764 Mon Sep  9 17:55:56 2013
new/usr/src/lib/libcndutils/common/uid.c
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License, Version 1.0 only
6  * (the "License"). You may not use this file except in compliance
7  * with the License.
8  *
9  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 * or http://www.opensolaris.org/os/licensing.
11 * See the License for the specific language governing permissions
12 * and limitations under the License.
13 *
14 * When distributing Covered Code, include this CDDL HEADER in each
15 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 * If applicable, add the following below this CDDL HEADER, with the
17 * fields enclosed by brackets "[]" replaced with your own identifying
18 * information: Portions Copyright [yyyy] [name of copyright owner]
19 *
20 * CDDL HEADER END
21 */
22 /*
23  * Copyright (c) 1997-2001 by Sun Microsystems, Inc.
24  * All rights reserved.
25 */

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

30 /*
31  * Copyright (c) 2013 RackTop Systems.
32  */
33 #endif /* ! codereview */

35 #include <errno.h>
36 #pragma ident "%Z%%M% %I%      %E% SMI" /* SVr4.0 1.5 */

36 #include <sys/types.h>
37 #include <stdio.h>
38 #include <userdefs.h>
39 #include <pwd.h>
40 #include <libcndutils.h>
41 #endif /* ! codereview */

43 static int findunuseduid(uid_t start, uid_t stop, uid_t *ret);
44 static boolean_t isreserveduid(uid_t uid);
45 #include <sys/param.h>
46 #ifndef MAXUID
47 #include <limits.h>
48 #endif
49 #ifndef UID_MAX
50 #define MAXUID UID_MAX
51 #else
52 #define MAXUID 60000
53 #endif
54 #endif

46 static uid_t getrangeboundid(uid_t start, uid_t stop);

```

```

47 static int isreserveduid(uid_t uid);

46 /*
47  * Find the highest unused uid. If the highest unused uid is "stop",
48  * then attempt to find a hole in the range. Returns 0 on success.
49  * Find the highest uid currently in use and return it. If the highest unused
50  * uid is MAXUID, then attempt to find a hole in the range. If there are no
51  * more unused uids, then return -1.
52  */
53 int
54 findnextuid(uid_t start, uid_t stop, uid_t *ret)
55 uid_t
56 findnextuid(void)
57 {
58     uid_t uid = start;
59     uid_t uid = DEFRID + 1;
60     struct passwd *pwd;
61     boolean_t overflow = B_FALSE;
62     uchar_t overflow = 0;

63     setpwent();
64     for (pwd = getpwent(); pwd != NULL; pwd = getpwent()) {
65         if (isreserveduid(pwd->pw_uid)) /* Skip reserved IDs */
66             continue;
67         if (pwd->pw_uid >= uid) {
68             if (pwd->pw_uid == stop) { /* Overflow check */
69                 overflow = B_TRUE;
70             }
71             if (pwd->pw_uid == MAXUID) { /* Overflow check */
72                 overflow = 1;
73                 break;
74             }
75             uid = pwd->pw_uid + 1;
76             while (isreserveduid(uid) &&
77                 uid < MAXUID) { /* Skip reserved IDs */
78                 uid++;
79             }
80         }
81     }
82     if (pwd == NULL && errno != 0) {
83         endpwent();
84         return (-1);
85     }
86 #endif /* ! codereview */
87     endpwent();
88     if (overflow == B_TRUE) /* Find a hole */
89         return (findunuseduid(start, stop, ret));
90     while (isreserveduid(uid) && uid < stop) /* Skip reserved IDs */
91         uid++;
92     *ret = uid;
93     return (0);
94     if (overflow == 1) /* Find a hole */
95         return (getrangeboundid(DEFRID + 1, MAXUID));
96     return (uid);
97 }

98 /*
99  * Check to see whether the uid is a reserved uid
100 * Check to see that the uid is a reserved uid
101 * -- nobody, noaccess or nobody4
102 */
103 static boolean_t
104 isreserveduid(uid_t uid)
105 {
106     return (uid == 60001 || uid == 60002 || uid == 65534);
107 }

```

```
93 /*
94 * findunuseduid() attempts to return the next valid usable id between the
95 * supplied upper and lower limits. Returns 0 on success.
96 * getrangeboundid() attempts to return the next valid usable id between the
97 * supplied upper and lower limits. If these limits exceed the system
98 * boundaries of DEFRID +1 and MAXUID (lower and upper bound respectively),
99 * then they are ignored and DEFRID + 1 and MAXUID are used.
100 */
101 static int
102 findunuseduid(uid_t start, uid_t stop, uid_t *ret)
103 static uid_t
104 getrangeboundid(uid_t start, uid_t stop)
105 {
106     uid_t low = (start <= DEFRID) ? DEFRID + 1 : start;
107     uid_t high = (stop < MAXUID) ? stop : MAXUID;
108     uid_t uid;
109
110     for (uid = start; uid <= stop; uid++) {
111         for (uid = low; uid <= high; uid++) {
112             if (isreserveduid(uid))
113                 continue;
114             if (getpwuid(uid) == NULL) {
115                 if (errno != 0)
116                     return (-1);
117                 if (getpwuid(uid) == NULL)
118                     break;
119             }
120         }
121     }
122     if (uid > stop)
123         return (-1);
124     *ret = uid;
125     return (0);
126     return ((uid > high) ? -1 : uid);
127 }
128
129 _____unchanged_portion_omitted_____
```

new/usr/src/lib/libcmdutils/libcmdutils.h

1

```
*****
3901 Mon Sep  9 17:55:58 2013
new/usr/src/lib/libcmdutils/libcmdutils.h
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
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 /*
26  * Copyright (c) 2013 RackTop Systems.
27  */
28 #endif /* ! codereview */

30 /*
31  * Declarations for the functions in libcmdutils.
32  */

34 #ifndef _LIBCMDUTILS_H
35 #define _LIBCMDUTILS_H

37 #pragma ident    "%Z%M% %I%    %E% SMI"

39 /*
40  * This is a private header file. Applications should not directly include
41  * this file.
42  */

44 #include <stdio.h>
45 #include <unistd.h>
46 #include <stdlib.h>
47 #include <errno.h>
48 #include <fcntl.h>
49 #include <limits.h>
50 #include <libintl.h>
51 #include <string.h>
52 #include <dirent.h>
53 #include <attr.h>
54 #include <sys/avl.h>
55 #include <sys/types.h>
56 #include <sys/stat.h>
57 #include <sys/mman.h>
```

new/usr/src/lib/libcmdutils/libcmdutils.h

2

```
58 #include <libnvpair.h>

60 #ifdef __cplusplus
61 extern "C" {
62 #endif

64 /* extended system attribute support */
65 #define _NOT_SATTR    0
66 #define _RO_SATTR    1
67 #define _RW_SATTR    2

69 #define MAXMAPSIZE    (1024*1024*8) /* map at most 8MB */
70 #define SMALLFILESIZE (32*1024) /* don't use mmap on little file */
71 #define ISREG(A)      (((A).st_mode & S_IFMT) == S_IFREG)

73 /* avltree */
74 #define OFFSETOF(s, m) (((size_t)&(((s *)0)->m)))

76 /* Type used for a node containing a device id and inode number */
77 typedef struct tree_node {
78     dev_t      node_dev;
79     ino_t      node_ino;
80     avl_node_t avl_link;
81 } tree_node_t;

84 /* extended system attribute support */

86 /* Determine if a file is the name of an extended system attribute file */
87 extern int sysattr_type(char *);

89 /* Determine if the underlying file system supports system attributes */
90 extern int sysattr_support(char *, int);

92 /* Copies the content of the source file to the target file */
93 extern int writefile(int, int, char *, char *, char *, char *,
94 struct stat *, struct stat *);

96 /* Gets file descriptors of the source and target attribute files */
97 extern int get_attrdirs(int, int, char *, int *, int *);

99 /* Move extended attribute and extended system attribute */
100 extern int mv_xattrs(char *, char *, char *, int, int);

102 /* Returns non default extended system attribute list */
103 extern nvlist_t *sysattr_list(char *, int, char *);

107 /* avltree */

109 /*
110  * Used to compare two nodes. We are attempting to match the 1st
111  * argument (node) against the 2nd argument (a node which
112  * is already in the search tree).
113  */

115 extern int tnode_compare(const void *, const void *);

117 /*
118  * Used to add a single node (containing the input device id and
119  * inode number) to the specified search tree. The calling
120  * application must set the tree pointer to NULL before calling
121  * add_tnode() for the first time.
122  */
123 extern int add_tnode(avl_tree_t **, dev_t, ino_t);
```

```
125 /*
126  * Used to destroy a whole tree (all nodes) without rebalancing.
127  * The calling application is responsible for setting the tree
128  * pointer to NULL upon return.
129  */
130 extern void destroy_tree(avl_tree_t *);

134          /* user/group id helpers */

136 /*
137  * Used to get the next available user id in given range.
138  */
139 extern int findnextuid(uid_t, uid_t, uid_t *);

141 /*
142  * Used to get the next available group id in given range.
143  */
144 extern int findnextgid(gid_t, gid_t, gid_t *);

146 #endif /* ! codereview */
147 #ifdef __cplusplus
148 }
149 #endif

151 #endif /* _LIBCMDUTILS_H */
```

```

*****
61286 Mon Sep  9 17:55:58 2013
new/usr/src/lib/smb/srv/lib/smb/common/smb_lgrp.c
4078 groupadd execs getent unnecessarily
Reviewed by: Rich Lowe <richlowe@richlowe.net>
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: Milan Jurik <milan.jurik@xylab.cz>
Reviewed by: Gordon Ross <Gordon.W.Ross@gmail.com>
*****
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 2011 Nexenta Systems, Inc. All rights reserved.
24  * Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
25  * Copyright (c) 2013 RackTop Systems.
26 #endif /* ! codereview */
27 */

29 #include <stdlib.h>
30 #include <strings.h>
31 #include <unistd.h>
32 #include <syslog.h>
33 #include <thread.h>
34 #include <synch.h>
35 #include <grp.h>
36 #include <assert.h>
37 #include <libintl.h>
38 #include <smb/srv/lib/smb.h>
39 #include <smb_sqlite.h>
40 #include <sys/types.h>
41 #include <sys/stat.h>
42 #include <sys/param.h>
43 #include <libcmdutils.h>
44 #endif /* ! codereview */

46 /*
47  * Local domain SID (aka machine SID) is not stored in the domain table
48  * therefore the index is 0
49  */
50 #define SMB_LGRP_LOCAL_IDX      0
51 #define SMB_LGRP_BUILTIN_IDX   1

53 #define SMB_LGRP_DB_NAME       "/var/smb/smbgroup.db"
54 #define SMB_LGRP_DB_TIMEOUT    3000      /* in millisecond */
55 #define SMB_LGRP_DB_VERMAJOR   1
56 #define SMB_LGRP_DB_VERMINOR  0
57 #define SMB_LGRP_DB_MAGIC      0x4C475250 /* LGRP */

```

```

59 #define SMB_LGRP_DB_ORD      1      /* open read-only */
60 #define SMB_LGRP_DB_ORW     2      /* open read/write */

62 #define SMB_LGRP_DB_ADDMEMBER 1
63 #define SMB_LGRP_DB_DELMEMBER 2

65 /*
66  * members column of the groups table is an array of
67  * member structure smb_lgmid_t defined below.
68  *
69  * privs column of the groups table is an array of bytes
70  * where each byte is the id of an enable privilege
71  */
72 #define SMB_LGRP_DB_SQL \
73     "CREATE TABLE db_info ("      \
74         "    ver_major INTEGER,"    \
75         "    ver_minor INTEGER,"    \
76         "    magic    INTEGER"     \
77     ");"                            \
78     ""                               \
79     "CREATE TABLE domains ("      \
80         "    dom_idx INTEGER PRIMARY KEY," \
81         "    dom_sid TEXT UNIQUE,"      \
82         "    dom_cnt INTEGER"         \
83     ");"                            \
84     ""                               \
85     "CREATE UNIQUE INDEX domsid_idx ON domains (dom_sid);" \
86     ""                               \
87     "CREATE TABLE groups ("       \
88         "    name    TEXT PRIMARY KEY," \
89         "    sid_idx INTEGER,"        \
90         "    sid_rid INTEGER,"       \
91         "    sid_type INTEGER,"     \
92         "    sid_attrs INTEGER,"    \
93         "    comment TEXT,"        \
94         "    n_privs INTEGER,"      \
95         "    privs  BLOB,"         \
96         "    n_members INTEGER,"   \
97         "    members BLOB"        \
98     ");"                            \
99     ""                               \
100    "CREATE INDEX grprid_idx ON groups (sid_rid);"

102 /*
103  * Number of groups table columns
104  */
105 #define SMB_LGRP_GTBL_NCOL      10

107 #define SMB_LGRP_GTBL_NAME      0
108 #define SMB_LGRP_GTBL_SIDIDX   1
109 #define SMB_LGRP_GTBL_SIDRID   2
110 #define SMB_LGRP_GTBL_SIDTYP   3
111 #define SMB_LGRP_GTBL_SIDATR   4
112 #define SMB_LGRP_GTBL_CMNT     5
113 #define SMB_LGRP_GTBL_NPRIVS   6
114 #define SMB_LGRP_GTBL_PRIVS    7
115 #define SMB_LGRP_GTBL_NMEMBS   8
116 #define SMB_LGRP_GTBL_MEMBS    9

118 #define SMB_LGRP_INFO_NONE     0x00
119 #define SMB_LGRP_INFO_NAME     0x01
120 #define SMB_LGRP_INFO_CMNT     0x02
121 #define SMB_LGRP_INFO_SID      0x04
122 #define SMB_LGRP_INFO_PRIV     0x08
123 #define SMB_LGRP_INFO_MEMB     0x10

```

```

124 #define SMB_LGRP_INFO_ALL      0x1F

126 #define SMB_LGRP_PGRP_GRPtmp  "/etc/gtmp"
127 #define SMB_LGRP_PGRP_GRPBUFSIZ 5120
128 #define SMB_LGRP_PGRP_GROUP    "/etc/group"
129 #define SMB_LGRP_PGRP_MAXGLEN  9      /* max length of group name */
130 #define SMB_LGRP_PGRP_DEFRID   1000  /* lowest cifs created gid */
131 #define SMB_LGRP_PGRP_DEFRID   99    /* max reserved id */

132 #define SMB_LGRP_PGRP_NOTUNIQUE 0
133 #define SMB_LGRP_PGRP_RESERVED 1
134 #define SMB_LGRP_PGRP_UNIQUE   2
135 #define SMB_LGRP_PGRP_TOOBIG   3
136 #define SMB_LGRP_PGRP_INVALID  4

138 #define NULL_MSGCHK(msg)      ((msg) ? (msg) : "NULL")

140 /* Member ID */
141 typedef struct smb_lgmid {
142     uint32_t m_idx;
143     uint32_t m_rid;
144     uint16_t m_type;
145 } smb_lgmid_t;
    unchanged portion omitted

2551 /*
2447 * smb_lgrp_pgrp_valid_gid
2448 *
2449 * Check to see that the gid is not a reserved gid
2450 * -- nobody (60001), noaccess (60002) or nogroup (65534)
2451 */
2452 static int
2453 smb_lgrp_pgrp_valid_gid(gid_t gid)
2454 {
2455     return (gid != 60001 && gid != 60002 && gid != 65534);
2456 }

2458 /*
2459 * smb_lgrp_pgrp_findnextgid(void)
2460 *
2461 * This method finds the next valid GID.
2462 * It sorts the used GIDs in decreasing order to return MAXUSED + 1.
2463 * It then adds one to obtain the next valid GID.
2464 * On failure, -1 is returned. On success, a valid GID is returned.
2465 */
2466 static int
2467 smb_lgrp_pgrp_findnextgid(void)
2468 {
2469     FILE *fptr;
2470     gid_t last, next;
2471     int gid;

2473     if ((fptr = popen("exec sh -c "
2474         "\"getent group/cut -f3 -d:/sort -nr/uniq \" 2>/dev/null",
2475         "r")) == NULL)
2476         return (-1);

2478     if (fscanf(fp, "%u\n", &next) == EOF) {
2479         (void) pclose(fp);
2480         return (SMB_LGRP_PGRP_DEFRID + 1);
2481     }

2483     last = MAXUID;
2484     gid = -1;
2485     do {
2486         if (!smb_lgrp_pgrp_valid_gid(next))

```

```

2487         continue;

2489         if (next <= SMB_LGRP_PGRP_DEFRID) {
2490             if (last != SMB_LGRP_PGRP_DEFRID + 1)
2491                 gid = SMB_LGRP_PGRP_DEFRID + 1;
2492             break;
2493         }

2495         if ((gid = next + 1) != last) {
2496             while (!smb_lgrp_pgrp_valid_gid((gid_t)gid))
2497                 gid++;
2498             if (gid > 0 && gid < last)
2499                 break;
2500         }

2502         gid = -1;
2503         last = next;
2504     } while (fscanf(fp, "%u\n", &next) != EOF);

2506     (void) pclose(fp);
2507     return (gid);
2508 }

2510 /*
2511 * smb_lgrp_pgrp_add
2512 *
2513 * Create a posix group with the given name.
2514 * This group will be added to the /etc/group file.
2515 */
2516 static int
2517 smb_lgrp_pgrp_add(char *group)
2518 {
2519     FILE *etcgrp;
2520     FILE *etctmp;
2521     int o_mask;
2522     int o_mask, gret;
2523     int newdone = 0;
2524     struct stat sb;
2525     char buf[SMB_LGRP_PGRP_GRPBUFSIZ];
2526     gid_t gid;
2527     int rc = 0;

2529     rc = smb_lgrp_pgrp_valid_gname(group);
2530     if ((rc == SMB_LGRP_PGRP_INVALID) || (rc == SMB_LGRP_PGRP_NOTUNIQUE))
2531         return (-1);

2533     if ((findnextgid(SMB_LGRP_PGRP_DEFRID, MAXUID, &gid)) != 0)
2534         if ((gret = smb_lgrp_pgrp_findnextgid()) < 0)
2535             return (-1);
2536     gid = gret;

2538     if ((etcgrp = fopen(SMB_LGRP_PGRP_GROUP, "r")) == NULL)
2539         return (-1);

2541     if (fstat(fileno(etcgrp), &sb) < 0)
2542         sb.st_mode = S_IRUSR|S_IWUSR|S_IRGRP|S_IROTH;

2544     o_mask = umask(077);
2545     etctmp = fopen(SMB_LGRP_PGRP_GRPtmp, "w+");
2546     (void) umask(o_mask);

2548     if (etctmp == NULL) {
2549         (void) fclose(etcgrp);
2550         return (-1);
2551     }

```

```
2591     if (lockf(fileno(etctmp), F_LOCK, 0) != 0) {
2592         (void) fclose(etcgrp);
2593         (void) fclose(etctmp);
2594         (void) unlink(SMB_LGRP_PGRP_GRPTEMP);
2595         return (-1);
2596     }
2598     if (fchmod(fileno(etctmp), sb.st_mode) != 0 ||
2599         fchown(fileno(etctmp), sb.st_uid, sb.st_gid) != 0) {
2600         (void) lockf(fileno(etctmp), F_ULOCK, 0);
2601         (void) fclose(etcgrp);
2602         (void) fclose(etctmp);
2603         (void) unlink(SMB_LGRP_PGRP_GRPTEMP);
2604         return (-1);
2605     }
2607     while (fgets(buf, SMB_LGRP_PGRP_GRPBUFSIZ, etcgrp) != NULL) {
2608         /* Check for NameService reference */
2609         if (!newdone && (buf[0] == '+' || buf[0] == '-')) {
2610             (void) fprintf(etctmp, "%s::%u:\n", group, gid);
2611             newdone = 1;
2612         }
2614         (void) fputs(buf, etctmp);
2615     }
2616     (void) fclose(etcgrp);
2618     if (!newdone)
2619         (void) fprintf(etctmp, "%s::%u:\n", group, gid);
2621     if (rename(SMB_LGRP_PGRP_GRPTEMP, SMB_LGRP_PGRP_GROUP) < 0) {
2622         (void) lockf(fileno(etctmp), F_ULOCK, 0);
2623         (void) fclose(etctmp);
2624         (void) unlink(SMB_LGRP_PGRP_GRPTEMP);
2625         return (-1);
2626     }
2628     (void) lockf(fileno(etctmp), F_ULOCK, 0);
2629     (void) fclose(etctmp);
2630     return (0);
2631 }
```

unchanged portion omitted