

## new/exception\_lists/check\_rtime

1

```

*****
9470 Wed May 7 18:40:46 2014
new/exception_lists/check_rtime
2837 - remove print/lp* from gate and use CUPS from userland
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 #
26 # This file provides exceptions to the usual rules applied to ELF objects by
27 # check_rtime. All strings are Perl regular expressions that are compared to
28 # file paths. In addition to the standard Perl syntax, there is one extension:
29 #
30 #     MACH(dir)
31 #
32 # is expanded into a regular expression that matches the given
33 # directory, or a 64-bit subdirectory of the directory with the
34 # name of a 64-bit architecture. For example, MACH(lib) will match
35 # any of the following:
36 #
37 #     lib
38 #     lib/amd64
39 #     lib/sparcv9
40 #
41 #
42 # Directory hierarchies to skip completely
43 SKIP ^usr/lib/libc/ # optimized libc
44 SKIP ^usr/lib/rcm/ # 4426119
45 SKIP ^usr/perl5/ # alan's taking care of these :-
46 SKIP ^usr/src/ # no objects in source code
47 #
48 # Individual files that we don't examine
49 SKIP ^boot/grub/bin/grub$
50 SKIP ^usr/apache2/2.2/libexec/mod_ipp.so$ # Apache loadable module
51 SKIP ^usr/apache/libexec/mod_ipp.so$ # Apache loadable module
52 # USIII specific extns. cause ldd noise on USII bld. m/c
53 SKIP ^usr/lib/fps/sun4u/UltraSPARC.*/*fptest$
54 SKIP ^usr/MACH(lib)/lddstub$ # lddstub has no dependencies
55 SKIP ^usr/MACH(lib)/libssagent\.\so\.\1$ # 4328854
56 SKIP ^usr/lib/MACH(iconv)/geniconvtbl.so$ # 4384329
57 #
58 #
59 #
60 #
61 #
62 #
63 #
64 #
65 #
66 #
67 #
68 #
69 #
70 #
71 #
72 #
73 #
74 #
75 #
76 #
77 #
78 #
79 #
80 #
81 #
82 #
83 #
84 #
85 #
86 #
87 #
88 #
89 #
90 #
91 #
92 #
93 #
94 #
95 #
96 #
97 #
98 #
99 #
100 #
101 #
102 #
103 #
104 #
105 #
106 #
107 #
108 #
109 #
110 #
111 #
112 #
113 #
114 #
115 #
116 #
117 #
118 #
119 #
120 #
121 #
122 #
123 #
124 #
125 #
126 #
127 #
128 #
129 #
130 #
131 #
132 #
133 #
134 #
135 #
136 #
137 #
138 #
139 #
140 #
141 #
142 #
143 #
144 #
145 #
146 #
147 #
148 #
149 #
150 #
151 #
152 #
153 #
154 #
155 #
156 #
157 #
158 #
159 #
160 #
161 #
162 #
163 #
164 #
165 #
166 #
167 #
168 #
169 #
170 #
171 #
172 #
173 #
174 #
175 #
176 #
177 #
178 #
179 #
180 #
181 #
182 #
183 #
184 #
185 #
186 #
187 #
188 #
189 #
190 #
191 #
192 #
193 #
194 #
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #
221 #
222 #
223 #
224 #
225 #
226 #
227 #
228 #
229 #
230 #
231 #
232 #
233 #
234 #
235 #
236 #
237 #
238 #
239 #
240 #
241 #
242 #
243 #
244 #
245 #
246 #
247 #
248 #
249 #
250 #
251 #
252 #
253 #
254 #
255 #
256 #
257 #
258 #
259 #
260 #
261 #
262 #
263 #
264 #
265 #
266 #
267 #
268 #
269 #
270 #
271 #
272 #
273 #
274 #
275 #
276 #
277 #
278 #
279 #
280 #
281 #
282 #
283 #
284 #
285 #
286 #
287 #
288 #
289 #
290 #
291 #
292 #
293 #
294 #
295 #
296 #
297 #
298 #
299 #
300 #
301 #
302 #
303 #
304 #
305 #
306 #
307 #
308 #
309 #
310 #
311 #
312 #
313 #
314 #
315 #
316 #
317 #
318 #
319 #
320 #
321 #
322 #
323 #
324 #
325 #
326 #
327 #
328 #
329 #
330 #
331 #
332 #
333 #
334 #
335 #
336 #
337 #
338 #
339 #
340 #
341 #
342 #
343 #
344 #
345 #
346 #
347 #
348 #
349 #
350 #
351 #
352 #
353 #
354 #
355 #
356 #
357 #
358 #
359 #
360 #
361 #
362 #
363 #
364 #
365 #
366 #
367 #
368 #
369 #
370 #
371 #
372 #
373 #
374 #
375 #
376 #
377 #
378 #
379 #
380 #
381 #
382 #
383 #
384 #
385 #
386 #
387 #
388 #
389 #
390 #
391 #
392 #
393 #
394 #
395 #
396 #
397 #
398 #
399 #
400 #
401 #
402 #
403 #
404 #
405 #
406 #
407 #
408 #
409 #
410 #
411 #
412 #
413 #
414 #
415 #
416 #
417 #
418 #
419 #
420 #
421 #
422 #
423 #
424 #
425 #
426 #
427 #
428 #
429 #
430 #
431 #
432 #
433 #
434 #
435 #
436 #
437 #
438 #
439 #
440 #
441 #
442 #
443 #
444 #
445 #
446 #
447 #
448 #
449 #
450 #
451 #
452 #
453 #
454 #
455 #
456 #
457 #
458 #
459 #
460 #
461 #
462 #
463 #
464 #
465 #
466 #
467 #
468 #
469 #
470 #
471 #
472 #
473 #
474 #
475 #
476 #
477 #
478 #
479 #
480 #
481 #
482 #
483 #
484 #
485 #
486 #
487 #
488 #
489 #
490 #
491 #
492 #
493 #
494 #
495 #
496 #
497 #
498 #
499 #
500 #
501 #
502 #
503 #
504 #
505 #
506 #
507 #
508 #
509 #
510 #
511 #
512 #
513 #
514 #
515 #
516 #
517 #
518 #
519 #
520 #
521 #
522 #
523 #
524 #
525 #
526 #
527 #
528 #
529 #
530 #
531 #
532 #
533 #
534 #
535 #
536 #
537 #
538 #
539 #
540 #
541 #
542 #
543 #
544 #
545 #
546 #
547 #
548 #
549 #
550 #
551 #
552 #
553 #
554 #
555 #
556 #
557 #
558 #
559 #
560 #
561 #
562 #
563 #
564 #
565 #
566 #
567 #
568 #
569 #
570 #
571 #
572 #
573 #
574 #
575 #
576 #
577 #
578 #
579 #
580 #
581 #
582 #
583 #
584 #
585 #
586 #
587 #
588 #
589 #
590 #
591 #
592 #
593 #
594 #
595 #
596 #
597 #
598 #
599 #
600 #
601 #
602 #
603 #
604 #
605 #
606 #
607 #
608 #
609 #
610 #
611 #
612 #
613 #
614 #
615 #
616 #
617 #
618 #
619 #
620 #
621 #
622 #
623 #
624 #
625 #
626 #
627 #
628 #
629 #
630 #
631 #
632 #
633 #
634 #
635 #
636 #
637 #
638 #
639 #
640 #
641 #
642 #
643 #
644 #
645 #
646 #
647 #
648 #
649 #
650 #
651 #
652 #
653 #
654 #
655 #
656 #
657 #
658 #
659 #
660 #
661 #
662 #
663 #
664 #
665 #
666 #
667 #
668 #
669 #
670 #
671 #
672 #
673 #
674 #
675 #
676 #
677 #
678 #
679 #
680 #
681 #
682 #
683 #
684 #
685 #
686 #
687 #
688 #
689 #
690 #
691 #
692 #
693 #
694 #
695 #
696 #
697 #
698 #
699 #
700 #
701 #
702 #
703 #
704 #
705 #
706 #
707 #
708 #
709 #
710 #
711 #
712 #
713 #
714 #
715 #
716 #
717 #
718 #
719 #
720 #
721 #
722 #
723 #
724 #
725 #
726 #
727 #
728 #
729 #
730 #
731 #
732 #
733 #
734 #
735 #
736 #
737 #
738 #
739 #
740 #
741 #
742 #
743 #
744 #
745 #
746 #
747 #
748 #
749 #
750 #
751 #
752 #
753 #
754 #
755 #
756 #
757 #
758 #
759 #
760 #
761 #
762 #
763 #
764 #
765 #
766 #
767 #
768 #
769 #
770 #
771 #
772 #
773 #
774 #
775 #
776 #
777 #
778 #
779 #
780 #
781 #
782 #
783 #
784 #
785 #
786 #
787 #
788 #
789 #
790 #
791 #
792 #
793 #
794 #
795 #
796 #
797 #
798 #
799 #
800 #
801 #
802 #
803 #
804 #
805 #
806 #
807 #
808 #
809 #
810 #
811 #
812 #
813 #
814 #
815 #
816 #
817 #
818 #
819 #
820 #
821 #
822 #
823 #
824 #
825 #
826 #
827 #
828 #
829 #
830 #
831 #
832 #
833 #
834 #
835 #
836 #
837 #
838 #
839 #
840 #
841 #
842 #
843 #
844 #
845 #
846 #
847 #
848 #
849 #
850 #
851 #
852 #
853 #
854 #
855 #
856 #
857 #
858 #
859 #
860 #
861 #
862 #
863 #
864 #
865 #
866 #
867 #
868 #
869 #
870 #
871 #
872 #
873 #
874 #
875 #
876 #
877 #
878 #
879 #
880 #
881 #
882 #
883 #
884 #
885 #
886 #
887 #
888 #
889 #
890 #
891 #
892 #
893 #
894 #
895 #
896 #
897 #
898 #
899 #
900 #
901 #
902 #
903 #
904 #
905 #
906 #
907 #
908 #
909 #
910 #
911 #
912 #
913 #
914 #
915 #
916 #
917 #
918 #
919 #
920 #
921 #
922 #
923 #
924 #
925 #
926 #
927 #
928 #
929 #
930 #
931 #
932 #
933 #
934 #
935 #
936 #
937 #
938 #
939 #
940 #
941 #
942 #
943 #
944 #
945 #
946 #
947 #
948 #
949 #
950 #
951 #
952 #
953 #
954 #
955 #
956 #
957 #
958 #
959 #
960 #
961 #
962 #
963 #
964 #
965 #
966 #
967 #
968 #
969 #
970 #
971 #
972 #
973 #
974 #
975 #
976 #
977 #
978 #
979 #
980 #
981 #
982 #
983 #
984 #
985 #
986 #
987 #
988 #
989 #
990 #
991 #
992 #
993 #
994 #
995 #
996 #
997 #
998 #
999 #
1000 #

```

## new/exception\_lists/check\_rtime

2

```

61 SKIP ^usr/lib/sysevent/modules/picl_slm.so$
62 #
63 # Objects that are allowed to have executable data segments
64 EXEC_DATA ^MACH(lib)/ld\.\so\.\1$
65 EXEC_DATA ^lib/libc\.\so\.\1$ # 6524709, 32-bit, needed for x86 only
66 EXEC_DATA ^lib/amd64/libumem\.\so\.\1$ # ptcumem
67 EXEC_DATA ^lib/libumem\.\so\.\1$ # ptcumem
68 EXEC_DATA ^opt/SUNWdtrt/tst/.\ustack/tst\.\helper\.\exe$
69 EXEC_DATA ^platform/./MACH(kernel)/unix$
70 EXEC_DATA ^platform/./multiboot$
71 #
72 # Objects that are allowed to have an executable stack
73 EXEC_STACK ^platform/./MACH(kernel)/unix$
74 EXEC_STACK ^platform/./multiboot$
75 #
76 # Objects for which we allow relocations to the text segment
77 TEXTREL ^platform/./MACH(kernel)/unix$
78 #
79 # Directories and files that are allowed to have no direct bound symbols
80 NODIRECT ^platform/./MACH(kernel)/unix$
81 NODIRECT ^usr/ucb
82 NODIRECT ^usr/4lib/sbcp$
83 #
84 # Identify any files that should be skipped when building a crle(1)
85 # configuration file. As the hwcap libraries can be loop-back mounted onto
86 # libc, these can confuse crle(1) because of their identical dev/inode.
87 NOCRLEALT ^usr/lib/libc/libc_hwcap[1-3]\.\so\.\1$
88 #
89 # Files that should contain debugging information.
90 STAB ^platform/./MACH(kernel)/unix$
91 #
92 # Files that are allowed undefined references
93 UNDEF_REF ^usr/lib/libnisdb\.\so\.\2$
94 UNDEF_REF ^usr/snadm/lib/libsvm\.\so\.\1$
95 #
96 # Objects allowed to have unused dependencies
97 UNUSED_DEPS ^usr/lib/picl/plugins/ # require devtree dependencies
98 UNUSED_DEPS ^usr/lib/libp # profile libc makes libm an unused dep of libc
99 #
100 # libm.so.2 dependency
101 UNUSED_OBJ unused object=.*MACH(libm)/libm_hwcap1\.\so\.\2
102 #
103 # Unused runpaths due to dlopen() use
104 UNUSED_RPATH /usr/lib/fs/autofs.*\ from\ .automountd
105 UNUSED_RPATH /etc/ppp/plugins.*\ from\ .pppd
106 UNUSED_RPATH /usr/lib/inet/ppp.*\ from\ .pppd
107 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libipseutil\.\so\.\1
108 UNUSED_RPATH /usr/platform/.*rsmlib.*\ from\ .*librsm\.\so\.\2
109 UNUSED_RPATH \${ORIGIN}.*\ from\ .*fcode.so
110 UNUSED_RPATH /opt/VRTSvxvm/lib.*\ from\ .*libdiskmgt\.\so\.\1
111 #
112 # Unused runpaths in picl code
113 UNUSED_RPATH /usr/platform/.*\ from\ .*/usr/platform
114 UNUSED_RPATH /usr/lib/picl/.*\ from\ .*/usr/platform
115 UNUSED_RPATH /usr/platform/.*\ from\ .*/usr/lib/picl
116 #
117 # Unused runpaths in non-OSNET objects we can't change
118 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libnss3\.\so
119 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libnssutil3\.\so
120 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libmime3\.\so
121 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libssl3\.\so
122 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libdbus-1\.\so\.\3
123 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libdbus-glib-1\.\so\.\2
124 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libglib-2\.\0\.\so\.\0
125 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libglib-2\.\0\.\so\.\0
126 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libobject-2\.\0\.\so\.\0

```

## new/exception\_lists/check\_rtime

3

```

127 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libgobject-2\.\0\.\so\.\0
128 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libgthread-2\.\0\.\so\.\0
129 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libgthread-2\.\0\.\so\.\0
130 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libcrypto\.\so\.\0\.\9\.\8
131 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libnetsnmp\.\so\.\15
132 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libgcc_s\.\so\.\1
133 UNUSED_RPATH /usr/ccs/lib.*\ from\ .*libgcc_s\.\so\.\1
134 UNUSED_RPATH /usr/lib.*\ from\ .*libgcc_s\.\so\.\1
135 UNUSED_RPATH /usr/postgres/8.3/lib.*\ from\ .*libpq\.\so\.\5
136 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libpq\.\so\.\5
137 UNUSED_RPATH /usr/lib.*\ from\ .*usr/lib/mps
138 UNUSED_RPATH /usr/ccs/lib.*\ from\ .*usr/lib/mps

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

```

## 145 # Unreferenced objects of non-OSnet objects we can't change

```

146 UNREF_OBJ /lib.*\ of\ .*libcimapi\.\so
147 UNREF_OBJ /lib.*\ of\ .*libdbus-1\.\so\.\3
148 UNREF_OBJ /lib.*\ of\ .*libdbus-glib-1\.\so\.\2
149 UNREF_OBJ /lib.*\ of\ .*libglib-2.0\.\so\.\0
150 UNREF_OBJ /lib.*\ of\ .*libgobject-2.0\.\so\.\0
151 UNREF_OBJ /lib.*\ of\ .*libgthread-2\.\0\.\so\.\0
152 UNREF_OBJ /lib.*\ of\ .*libjvm\.\so
153 UNREF_OBJ /lib.*\ of\ .*libnetsnmp\.\so\.\15
154 UNREF_OBJ /lib.*\ of\ .*libnetsnmpagent\.\so\.\15
155 UNREF_OBJ /lib.*\ of\ .*libnetsnmpmibs\.\so\.\15
156 UNREF_OBJ /lib.*\ of\ .*libnetsnmphelpers\.\so\.\15
157 UNREF_OBJ /lib.*\ of\ .*libnspr4\.\so
158 UNREF_OBJ /lib.*\ of\ .*libpq\.\so\.\5
159 UNREF_OBJ /lib.*\ of\ .*libsoftokn3\.\so
160 UNREF_OBJ /lib.*\ of\ .*libspmicommon\.\so\.\1
161 UNREF_OBJ /lib.*\ of\ .*libspmocommon\.\so\.\1
162 UNREF_OBJ /lib.*\ of\ .*libssl3\.\so
163 UNREF_OBJ /lib.*\ of\ .*libtspi\.\so\.\1
164 UNREF_OBJ /lib.*\ of\ .*libxml2\.\so\.\2
165 UNREF_OBJ /lib.*\ of\ .*libxslt\.\so\.\1
166 UNREF_OBJ /lib.*\ of\ .*libpq\.\so\.\4
167 UNREF_OBJ /lib.*\ of\ .*libpython2\.\4\.\so\.\1\.\0
168 UNREF_OBJ /libgcc_s.*\ of\ .*libstdc\+\+\.\so\.\6

```

## 170 # Unreferenced object of objects we can't change for other reasons

```

171 UNREF_OBJ /libmapmalloc\.\so\.\1;\ unused\ dependency\ of # interposer
172 UNREF_OBJ /libstdc\+\+\.\so\.\6;\ unused\ dependency\ of # gcc build
173 UNREF_OBJ /libm\.\so\.\2.*\ of\ .*libstdc\+\+\.\so\.\6 # gcc build
174 UNREF_OBJ /lib.*\ of\ .*lib/picl/plugins/ # picl
175 UNREF_OBJ /lib.*\ of\ .*kcfid # interposer
176 UNREF_OBJ /libpkcs11\.\so\.\1; .*\ of\ .*libkmf\.\so\.\1 # interposed
177 # Referenced by the Studio build, not the GCC build. GCC eliminates the unused
178 # statics which have the dependence.
179 UNREF_OBJ /libc\.\so\.\1.*\ of\ .*kldap\.\so\.\1

```

## 182 # Objects that used to contain system functionality that has since migrated to libc. We preserve these libraries as pure filters for backward compatibility but nothing needs to link to them.

```

185 OLDDEP libaio\.\so\.\1 # onnv build 44
186 OLDDEP libdl\.\so\.\1 # on10 build 49
187 OLDDEP libdoor\.\so\.\1 # onnv build 12
188 OLDDEP libintl\.\so\.\1 # on297 build 7
189 OLDDEP libpthread\.\so\.\1 # on10 build 53
190 OLDDEP librt\.\so\.\1 # onnv build 44
191 OLDDEP libsched\.\so\.\1 # on10 build 36
192 OLDDEP libthread\.\so\.\1 # on10 build 53

```

## new/exception\_lists/check\_rtime

4

```

193 OLDDEP libw\.\so\.\1 # on297 build 7

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

```

## 218 # The libprtdiag\_psr.so.1 objects built under usr/src/lib/libprtdiag\_psr

```

219 # are a family, all built using the same makefile, targeted at different
220 # sparc hardware variants. There are a small number of cases where this
221 # one size fits all approach causes an object to be linked against an
222 # unneeded library.
223 UNREF_OBJ lib/(libdevinfo|libcfgadm)\.\so\.\1; .*\ of\ .*SUNW,Netra-CP2300/1

```

```

*****
3913 Wed May 7 18:40:47 2014
new/exception_lists/interface_check
2837 - remove print/lp* from gate and use CUPS from userland
*****
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 #

23 # Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights reserved.

25 # This file provides exceptions to the usual rules applied to shared
26 # objects by interface_check. All strings are Perl regular expressions
27 # that are compared to file names. In addition to the standard Perl
28 # syntax, there is one extension:
29 #
30 #     MACH(dir)
31 #
32 # is expanded into a regular expression that matches the given
33 # directory, or a 64-bit subdirectory of the directory with the
34 # name of a 64-bit architecture. For example, MACH(lib) will match
35 # any of the following:
36 #
37 #     lib
38 #     lib/amd64
39 #     lib/sparcv9

42 # Shared objects underneath these parts of the tree are taken to be plugins.
43 # Plugins are not required to have versioned file names, and are not required
44 # to be internally versioned.
45 #
46 PLUGIN      ^usr/apache2/2.2/libexec
46 PLUGIN      ^usr/apache/libexec
47 PLUGIN      ^usr/lib/devfsadm
48 PLUGIN      ^usr/lib/efcode/*.so$
49 PLUGIN      ^usr/lib/elfedit
50 PLUGIN      ^usr/lib/fm/fmd/plugins
51 PLUGIN      ^usr/lib/fm/fmd/schemes
52 PLUGIN      ^usr/lib/fm/topo/plugins
53 PLUGIN      ^usr/lib/fwflash
54 PLUGIN      ^usr/lib/iconv
55 PLUGIN      ^usr/lib/inet/ppp
56 PLUGIN      ^usr/lib/mdb
57 PLUGIN      ^usr/lib/pci
58 PLUGIN      ^usr/lib/picl/plugins
59 PLUGIN      ^usr/lib/python2.[46]
60 PLUGIN      ^usr/lib/rcm/modules

```

```

61 PLUGIN      ^usr/lib/scsi/plugins
62 PLUGIN      ^usr/lib/sysevent/modules
63 PLUGIN      ^usr/perl5/5.[^\]*/*lib
64 PLUGIN      ^usr/platform
65 PLUGIN      ^usr/sadm/lib/wbem
66 # We unfortunately can't use MACH() here, since ../64/ is literal, and not a
67 # link to to amd64 or sparcv9
68 PLUGIN      ^usr/lib/dtrace/libdtrace_forceload\.$
69 PLUGIN      ^usr/lib/dtrace/64/libdtrace_forceload\.$

71 # sbcp is a special case, and not a plugin. However, it does not have a
72 # versioned name, and does not contain versioning, so the PLUGIN exemptions fit.
73 PLUGIN      ^usr/4lib/sbcp$

76 # Objects that are not expected to contain versioning information.
77 # Note that PLUGIN objects are automatically exempt from this,
78 # so these directives are generally applied to non-plugin objects
79 NOVERDEF    ^usr/4lib/libc\.$
80 NOVERDEF    ^usr/MACH(lib)/0@0\.$
81 NOVERDEF    ^usr/lib/MACH(abi)/appttrace\.$
82 NOVERDEF    ^usr/MACH(lib)/libfru.*\.$
83 NOVERDEF    ^usr/MACH(lib)/libkrb5\.$
84 NOVERDEF    ^usr/MACH(lib)/libzpool\.$
85 NOVERDEF    ^usr/MACH(lib)/madv\.$
86 NOVERDEF    ^usr/MACH(lib)/mpss\.$
87 NOVERDEF    ^usr/MACH(lib)/s10_brand\.$
88 NOVERDEF    ^usr/MACH(lib)/s10_npreload\.$
89 NOVERDEF    ^usr/MACH(lib)/snl_brand\.$
90 NOVERDEF    ^usr/lib/fs/{^}/fstyp\.$
91 NOVERDEF    ^usr/lib/libmilter\.$
92 NOVERDEF    ^usr/lib/libwrap\.$
93 NOVERDEF    ^usr/lib/locale/MACH(iso_8859_1)/iso_8859_1\.$
94 NOVERDEF    ^usr/lib/picl/plugins$
95 NOVERDEF    ^usr/sadm/admin/dhccpmgr/dhccpmgr\.$
96 NOVERDEF    ^usr/sadm/admin/printmgr/lib/libpmgr\.$

99 # Objects that are allowed to deviate from our standard version
100 # names.
101 NONSTD_VERNAME ^usr/MACH(lib)/libtecla\.$

104 # These libc variants have an SONAME of libc\.$
105 NONSTD_VERNAME ^usr/MACH(lib)/libc/libc_hwcap[1-3]+\.$

108 # The ABI requires the SONAME for libsys.so.1 to be /usr/lib/ld.so.1
109 # That means that the base version will also be /usr/lib/ld.so.1, which
110 # is non-standard.
111 NONSTD_VERNAME ^usr/lib/libsys\.$

```

```

*****
10285 Wed May 7 18:40:47 2014
new/usr/src/cmd/logadm/glob.c
2837 - remove print/lp* from gate and use CUPS from userland
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22 * Copyright 2006 Sun Microsystems, Inc. All rights reserved.
23 * Use is subject to license terms.
24 *
25 * logadm/glob.c -- globbing routines
26 *
27 * these routines support two kinds of globs. first, the
28 * usual kind of filename globbing, like:
29 *
30 *      *.c
31 *      /var/log/syslog.?
32 *      log[0-9]*file
33 *      /var/apache2/2.2/logs/x*{access,error}_log
34 *      /var/apache/logs/x*{access,error}_log
35 *
36 * this is basically the same syntax that csh supports for globs and
37 * is provided by the routine glob_glob() which takes a filename and
38 * returns a list of filenames that match the glob.
39 *
40 * the second type is something called a "reglob" which is a pathname
41 * where the components are regular expressions as described in regex(3c).
42 * some examples:
43 *
44 *      .*\.c
45 *      /var/log/syslog\..
46 *      log[0-9].*file
47 *      /var/log/syslog\.([0-9]+)$0
48 *
49 * the last example uses the ()$n form to assign a numeric extension
50 * on a filename to the "n" value kept by the fn routines with each
51 * filename (see fn_setn() in fn.c). logadm uses this mechanism to
52 * correctly sort lognames when templates containing $n are used.
53 *
54 * the routine glob_reglob() is used to expand reglobs. glob_glob()
55 * is implemented by expanding the curly braces, converting the globs
56 * to reglobs, and then passing the work to glob_reglob().
57 *
58 * finally, since expanding globs and reglobs requires doing a stat(2)
59 * on the files, we store the resulting stat information in the filename
60 * struct (see fn_setstat() in fn.c).

```

```

61 * the glob(3c) routines are not used here since they don't support
62 * braces, and don't support the more powerful reglobs required by logadm.
63 */
64
65 #pragma ident      "%Z%M% %I%      %E% SMI"
66
67 #include <stdio.h>
68 #include <libintl.h>
69 #include <stdlib.h>
70 #include <string.h>
71 #include <sys/types.h>
72 #include <sys/param.h>
73 #include <sys/stat.h>
74 #include <dirent.h>
75 #include "err.h"
76 #include "fn.h"
77 #include "glob.h"
78
79 /* forward declarations for functions used internally by this module */
80 static struct fn_list *glob_debrace(struct fn *fnp);
81 static struct fn_list *glob_reglob_list(struct fn_list *fnlp);
82 static boolean_t glob_magic(struct fn *fnp);
83
84 /* expand curly braces (like file{one,two,three}name) */
85 static struct fn_list *
86 glob_debrace(struct fn *fnp)
87 {
88     struct fn_list *ret = fn_list_new(NULL);
89     struct fn_list *newret;
90     char *sp = fn_s(fnp);
91     char *left;
92     char *right;
93     char *comma;
94
95     /* start with an empty string in the list */
96     fn_list_adds(ret, "");
97
98     /* while braces remain... */
99     while (sp != NULL && (left = strchr(sp, '{') != NULL)
100           && (right = strchr(left, '}') != NULL) {
101         err(EF_FILE|EF_JMP, "Missing ");
102         fn_list_free(ret);
103         return (NULL);
104     } else {
105         /* stuff before "left" is finished */
106         fn_list_appendrange(ret, sp, left);
107
108         /* stuff after "right" still need processing */
109         sp = right + 1;
110
111         if (left + 1 == right)
112             continue; /* just an empty {} */
113
114         /* stuff between "left" and "right" is comma-sep list */
115         left++;
116         newret = fn_list_new(NULL);
117         while ((comma = strchr(left, ',')) != NULL) {
118             struct fn_list *dup = fn_list_dup(ret);
119
120             /* stuff from left to comma is one variant */
121             fn_list_appendrange(dup, left, comma);
122             fn_list_addfn_list(newret, dup);
123             left = comma + 1;
124         }
125     }
126
127     /* what's left is the last item in the list */

```

new/usr/src/cmd/logadm/glob.c

3

```
125             fn_list_appendrange(ret, left, right);
126             fn_list_addfn_list(newret, ret);
127             ret = newret;
128         }
129
130     /* anything remaining in "s" is finished */
131     fn_list_appendrange(ret, sp, &sp[strlen(sp)]);
132     return (ret);
133 }
_____unchanged_portion_omitted_____
```

```

*****
36813 Wed May 7 18:40:47 2014
new/usr/src/cmd/logadm/main.c
2837 - remove print/lp* from gate and use CUPS from userland
*****
_____unchanged_portion_omitted_____

567 /* absurdly long buffer lengths for holding user/group/mode strings */
568 #define TimestRMAX 100
569 #define MAXATTR 100

571 /* rotate a log file if necessary, returns true if ok to go on to expire step */
572 static boolean_t
573 rotatelog(struct fn *fnp, struct opts *opts)
574 {
575     char *fname = fn_s(fnp);
576     struct stat stbuf;
577     char nowstr[TimestRMAX];
578     struct fn *recentlog = fn_new(NULL); /* for -R cmd */
579     char ownerbuf[MAXATTR];
580     char groupbuf[MAXATTR];
581     char modebuf[MAXATTR];
582     const char *owner;
583     const char *group;
584     const char *mode;

586     if (Debug && fname != NULL)
587         (void) fprintf(stderr, "rotatelog: fname <%s>\n", fname);

589     if (opts_count(opts, "p") && opts_optarg_int(opts, "p") == OPTP_NEVER)
590         return (B_TRUE); /* "-p never" forced no rotate */

592     /* prepare the keywords */
593     kw_init(fnp, NULL);
594     if (Debug > 1) {
595         (void) fprintf(stderr, "rotatelog keywords:\n");
596         kw_print(stderr);
597     }

599     if (lstat(fname, &stbuf) < 0) {
600         if (opts_count(opts, "N"))
601             return (1);
602         err(EF_WARN|EF_SYS, "%s", fname);
603         return (B_FALSE);
604     }

606     if ((stbuf.st_mode & S_IFMT) == S_IFLNK) {
607         err(EF_WARN, "%s is a symlink", fname);
608         return (B_FALSE);
609     }

611     if ((stbuf.st_mode & S_IFMT) != S_IFREG) {
612         err(EF_WARN, "%s is not a regular file", fname);
613         return (B_FALSE);
614     }

616     /* even if size condition is not met, this entry is "done" */
617     if (opts_count(opts, "s") &&
618         stbuf.st_size < opts_optarg_int(opts, "s")) {
619         Donenames = lut_add(Donenames, fname, "1");
620         return (B_TRUE);
621     }

623     /* see if age condition is present, and return if not met */
624     if (opts_count(opts, "p")) {

```

```

625         off_t when = opts_optarg_int(opts, "p");
626         struct opts *cfopts;

628         /* unless rotate forced by "-p now", see if period has passed */
629         if (when != OPTP_NOW) {
630             /*
631              * "when" holds the number of seconds that must have
632              * passed since the last time this log was rotated.
633              * of course, running logadm can take a little time
634              * (typically a second or two, but longer if the
635              * conf file has lots of stuff in it) and that amount
636              * of time is variable, depending on system load, etc.
637              * so we want to allow a little "slop" in the value of
638              * "when". this way, if a log should be rotated every
639              * week, and the number of seconds passed is really a
640              * few seconds short of a week, we'll go ahead and
641              * rotate the log as expected.
642              */
643             if (when >= 60 * 60)
644                 when -= 59;

647             /*
648              * last rotation is recorded as argument to -P,
649              * but if logname isn't the same as log file name
650              * then the timestamp would be recorded on a
651              * separate line in the timestamp file. so if we
652              * haven't seen a -P already, we check to see if
653              * it is part of a specific entry for the log
654              * file name. this handles the case where the
655              * logname is "apache", it supplies a log file
656              * name like "/var/apache2/2.2/logs/[a-z]*.log",
657              * which expands to multiple file names. if one of
658              * the file names is "/var/apache2/2.2/logs/access_log"
659              * name like "/var/apache/logs/[a-z]*.log",
660              * which expands to multiple file names. if one
661              * of the file names is "/var/apache/logs/access_log"
662              * the the -P will be attached to a line with that
663              * logname in the timestamp file.
664              */
665             if (opts_count(opts, "P")) {
666                 off_t last = opts_optarg_int(opts, "P");

667                 /* return if not enough time has passed */
668                 if (Now - last < when)
669                     return (B_TRUE);
670             } else if ((cfopts = conf_opts(fname)) != NULL &&
671                 opts_count(cfopts, "P")) {
672                 off_t last = opts_optarg_int(cfopts, "P");

673                 /*
674                  * just checking this means this entry
675                  * is now "done" if we're going through
676                  * the entire conf file
677                  */
678                 Donenames = lut_add(Donenames, fname, "1");

679                 /* return if not enough time has passed */
680                 if (Now - last < when)
681                     return (B_TRUE);
682             }
683         }
684     }

686     if (Debug)
687         (void) fprintf(stderr, "rotatelog: conditions met\n");

```

```

688     if (opts_count(opts, "l")) {
689         /* Change the time zone to local time zone */
690         if (putenv("TZ="))
691             err(EF_SYS, "putenv TZ");
692         tzset();
693         Now = time(0);
694
695         /* rename the log file */
696         rotateto(fnp, opts, 0, recentlog, B_FALSE);
697
698         /* Change the time zone to UTC */
699         if (putenv("TZ=UTC"))
700             err(EF_SYS, "putenv TZ");
701         tzset();
702         Now = time(0);
703     } else {
704         /* rename the log file */
705         rotateto(fnp, opts, 0, recentlog, B_FALSE);
706     }
707
708     /* determine owner, group, mode for empty log file */
709     if (opts_count(opts, "o"))
710         (void) strcpy(ownerbuf, opts_optarg(opts, "o"), MAXATTR);
711     else {
712         (void) snprintf(ownerbuf, MAXATTR, "%ld", stbuf.st_uid);
713     }
714     owner = ownerbuf;
715     if (opts_count(opts, "g"))
716         group = opts_optarg(opts, "g");
717     else {
718         (void) snprintf(groupbuf, MAXATTR, "%ld", stbuf.st_gid);
719         group = groupbuf;
720     }
721     (void) strcat(ownerbuf, ":", MAXATTR - strlen(ownerbuf));
722     (void) strcat(ownerbuf, group, MAXATTR - strlen(ownerbuf));
723     if (opts_count(opts, "m"))
724         mode = opts_optarg(opts, "m");
725     else {
726         (void) snprintf(modebuf, MAXATTR,
727             "%03lo", stbuf.st_mode & 0777);
728         mode = modebuf;
729     }
730
731     /* create the empty log file */
732     docmd(opts, NULL, Touch, fname, NULL, NULL);
733     docmd(opts, NULL, Chown, owner, fname, NULL);
734     docmd(opts, NULL, Chmod, mode, fname, NULL);
735
736     /* execute post-rotation command */
737     if (opts_count(opts, "R")) {
738         struct fn *rawcmd = fn_new(opts_optarg(opts, "R"));
739         struct fn *cmd = fn_new(NULL);
740
741         kw_init(recentlog, NULL);
742         (void) kw_expand(rawcmd, cmd, 0, B_FALSE);
743         docmd(opts, "-R cmd", Sh, "-c", fn_s(cmd), NULL);
744         fn_free(rawcmd);
745         fn_free(cmd);
746     }
747     fn_free(recentlog);
748
749     /*
750     * add "after" command to list of after commands. we also record
751     * the email address, if any, where the error output of the after
752     * command should be sent. if the after command is already on
753     * our list, add the email addr to the list the email addrs for

```

```

754     * that command (the after command will only be executed once,
755     * so the error output gets mailed to every address we've come
756     * across associated with this command).
757     */
758     if (opts_count(opts, "a")) {
759         const char *cmd = opts_optarg(opts, "a");
760         struct lut *addrs = (struct lut *)lut_lookup(Aftercmds, cmd);
761         if (opts_count(opts, "e"))
762             addrs = lut_add(addrs, opts_optarg(opts, "e"), NULL);
763         Aftercmds = lut_add(Aftercmds, opts_optarg(opts, "a"), addrs);
764     }
765
766     /* record the rotation date */
767     (void) strftime(nowstr, sizeof(nowstr),
768         "%a %b %e %T %Y", gmtime(&Now));
769     if (opts_count(opts, "v") && fname != NULL)
770         (void) out("# recording rotation date %s for %s\n",
771             nowstr, fname);
772     conf_set(fname, "P", STRDUP(nowstr));
773     Donenames = lut_add(Donenames, fname, "1");
774     return (B_TRUE);
775 }

```

---

unchanged portion omitted

new/usr/src/cmd/logadm/tester

1

```
*****
69921 Wed May 7 18:40:48 2014
new/usr/src/cmd/logadm/tester
2837 - remove print/lp* from gate and use CUPS from userland
*****
_____unchanged_portion_omitted_____
```

```
257 #####
258 #####
259 #####
260 #####
261 #####
```

THE TESTS START AFTER HERE...

```
263 # common setup step -- create a testfile.conf
264 sub set_testconffile {
265     my $fname = shift;
266     $fname = 'testfile.conf' unless defined($fname);
268     set_file($fname, <<'EOF');
269 #
270 # logadm.conf
271 #
272 # Default settings for system log file management.
273 # The -w option to logadm(1M) is the preferred way to write to this file,
274 # but if you do edit it by hand, use "logadm -V" to check it for errors.
275 #
276 # The format of lines in this file is:
277 #     <logname> <options>
278 # For each logname listed here, the default options to logadm
279 # are given. Options given on the logadm command line override
280 # the defaults contained in this file.
281 #
282 # logadm typically runs early every morning via an entry in
283 # root's crontab (see crontab(1)).
284 #
285 /var/adm/messages -C 4 -P 'Thu Nov 1 16:56:42 2001' -a 'kill -HUP `cat /var/run
286 /var/cron/log -s 512k -t /var/cron/olog
287 /var/lp/logs/lpsched -C 2 -N -t '$file.$N'
288 #
289 # The entry below is used by turnacct(1M)
290 #
291 /var/adm/pacct -C 0 -a '/usr/lib/acct/accton pacct' -g adm -m 664 -o adm -p neve
292 apache -C 24 -a '/usr/apache2/2.2/bin/apachectl graceful' -p lm -t '/var/apache2
293 /var/apache2/2.2/logs/access_log -P 'Thu Nov 1 08:27:56 2001'
294 /var/apache2/2.2/logs/error_log -P 'Thu Nov 1 08:27:56 2001'
295 /var/apache2/2.2/logs/suexec_log -P 'Thu Nov 1 08:27:56 2001'
296 /var/apache2/2.2/logs/mod_jserv.log -P 'Thu Nov 1 08:27:56 2001'
297 /var/apache2/2.2/logs/jserv.log -P 'Thu Nov 1 08:27:56 2001'
298 apache -C 24 -a '/usr/apache/bin/apachectl graceful' -p lm -t '/var/apache/old-1
299 /var/log/syslog -C 8 -P 'Thu Nov 1 09:16:38 2001' -a 'kill -HUP `cat /var/run/*
300 /var/apache/logs/access_log -P 'Thu Nov 1 08:27:56 2001'
301 /var/apache/logs/error_log -P 'Thu Nov 1 08:27:56 2001'
302 /var/apache/logs/suexec_log -P 'Thu Nov 1 08:27:56 2001'
303 /var/apache/logs/mod_jserv.log -P 'Thu Nov 1 08:27:56 2001'
304 /var/apache/logs/jserv.log -P 'Thu Nov 1 08:27:56 2001'
305 EOF
306 }
_____unchanged_portion_omitted_____
```

```
812 #####
813 #
814 #     logadmV1 -- test of "logadm -V"
815 #
816 #####
817 sub logadmV1 {
818     set_testconffile;
```

new/usr/src/cmd/logadm/tester

2

```
820     set_file('std.out.expect', <<'EOF');
821 /var/adm/messages -C 4 -P 'Thu Nov 1 16:56:42 2001' -a 'kill -HUP `cat /var/run
822 /var/cron/log -s 512k -t /var/cron/olog
823 /var/lp/logs/lpsched -C 2 -N -t '$file.$N'
824 /var/adm/pacct -C 0 -a '/usr/lib/acct/accton pacct' -g adm -m 664 -o adm -p neve
825 apache -C 24 -a '/usr/apache2/2.2/bin/apachectl graceful' -p lm -t '/var/apache2
826 apache -C 24 -a '/usr/apache/bin/apachectl graceful' -p lm -t '/var/apache/old-1
827 /var/log/syslog -C 8 -P 'Thu Nov 1 09:16:38 2001' -a 'kill -HUP `cat /var/run/*
828 /var/apache2/2.2/logs/access_log -P 'Thu Nov 1 08:27:56 2001'
829 /var/apache2/2.2/logs/error_log -P 'Thu Nov 1 08:27:56 2001'
830 /var/apache2/2.2/logs/suexec_log -P 'Thu Nov 1 08:27:56 2001'
831 /var/apache2/2.2/logs/mod_jserv.log -P 'Thu Nov 1 08:27:56 2001'
832 /var/apache2/2.2/logs/jserv.log -P 'Thu Nov 1 08:27:56 2001'
833 EOF
```

```
834     set_file('checktest', <<'EOF');
835 [ -s std.err ] && { cat std.err; exit 1; }
836 exec /bin/diff std.out.expect std.out
837 EOF
```

```
839     set_file('runtest', <<"EOF");
840 # test "logadmV1"
841 $envsetup
842 exec $bindir/logadm -f testfile.conf -F testfile.conf -V >std.out 2>std.err
843 EOF
844 }
_____unchanged_portion_omitted_____
```



new/usr/src/cmd/lp/cmd/lpsched/print-svc

1

```
*****
4199 Wed May 7 18:40:48 2014
new/usr/src/cmd/lp/cmd/lpsched/print-svc
2837 - remove print/lp* from gate and use CUPS from userland
*****
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #

28 . /lib/svc/share/smf_include.sh
29 . /lib/svc/share/ipf_include.sh

31 # SERVICE = parent service name
32 SERVICE='echo $SMF_FMRI | /usr/bin/cut -f1,2 -d':"

34 case "$1" in
35 'start')

37 isoopts='/usr/sbin/svccfg <<-EOF
38     select $SMF_FMRI
39     listpg cmd_opts

41     EOF'

43 if [ "$isopts" ]; then

45 #called by /usr/lib/lpsched; use cmd_opts properties only

47     num_notifiers='/bin/svcprop -p cmd_opts/num_notifiers $SMF_FMRI'
49     if [ "$num_notifiers" != "" ]; then
50         OPTS="$OPTS -n $num_notifiers"
51     fi

53     num_filters='/bin/svcprop -p cmd_opts/num_filters $SMF_FMRI'
55     if [ "$num_filters" != "" ]; then
56         OPTS="$OPTS -f $num_filters"
57     fi

59     fd_limit='/bin/svcprop -p cmd_opts/fd_limit $SMF_FMRI'
61     if [ "$fd_limit" != "" ]; then
```

new/usr/src/cmd/lp/cmd/lpsched/print-svc

2

```
62     OPTS="$OPTS -p $fd_limit"
63     fi

65     reserved_fds='/bin/svcprop -p cmd_opts/reserved_fds $SMF_FMRI'

67     if [ "$reserved_fds" != "" ]; then
68         OPTS="$OPTS -r $reserved_fds"
69     fi

71 # clear out cmd_opts property group

73     svccfg <<-EOF
74     select $SMF_FMRI
75     delpg cmd_opts

77     EOF

79 else

81 # We are here through enable; use lpsched properties
82 # Check for saved properties

84     num_notifiers='/bin/svcprop -p lpsched/num_notifiers $SERVICE'
85     if [ "$num_notifiers" != "" ] && [ "$num_notifiers" != "0" ]; then
86         OPTS="$OPTS -n $num_notifiers"
87     fi

89     num_filters='/bin/svcprop -p lpsched/num_filters $SERVICE'
90     if [ "$num_filters" != "" ] && [ "$num_filters" != "0" ]; then
91         OPTS="$OPTS -f $num_filters"
92     fi

94     fd_limit='/bin/svcprop -p lpsched/fd_limit $SERVICE'
95     if [ "$fd_limit" != "" ] && [ "$fd_limit" != "0" ]; then
96         OPTS="$OPTS -p $fd_limit"
97     fi

99     reserved_fds='/bin/svcprop -p lpsched/reserved_fds $SERVICE'
100    if [ "$reserved_fds" != "" ] && [ "$reserved_fds" != "0" ]; then
101        OPTS="$OPTS -r $reserved_fds"
102    fi
103 fi

105 # set temporary or permanent properties from OPTS

107     [ -f /usr/lib/lp/local/lpsched ] || exit $SMF_EXIT_ERR_CONFIG

109     /usr/lib/lp/local/lpsched ${OPTS}

111     ;;

113 'stop')
114     [ -f /usr/lib/lp/local/lpshut ] || exit $SMF_EXIT_ERR_CONFIG

116     /usr/lib/lp/local/lpshut
117     ;;

119 'ipfilter')
120     FMRI=$2
121     IPP_FMRI="svc:/application/print/ipp-listener:default"
122     RFC1179_FMRI="svc:/application/print/rfc1179:default"
123     IPP_CONF=/etc/apache2/2.2/httpd-standalone-ipp.conf
124     IPP_CONF=/etc/apache/httpd-standalone-ipp.conf
125     ip="any"

126     policy='get_policy $FMRI'
```

```
128     file=`fmri_to_file $RFC1179_FMRI $IPF_SUFFIX`
129     echo "# $RFC1179_FMRI" >$file
130     service_is_enabled ${RFC1179_FMRI}
131     if [ $? -eq 0 ]; then
132         rfc_name='svcprop -p inetd/name ${RFC1179_FMRI} 2>/dev/null`
133         rfc_proto='svcprop -p inetd/proto ${RFC1179_FMRI} 2>/dev/null |
134             sed 's/6/ /`
135         rfc_port='$$SERVINFO -p -t -s $rfc_name`
136         generate_rules $FMRI $policy $rfc_proto $ip $rfc_port $file
137     fi

139     file=`fmri_to_file $IPP_FMRI $IPF_SUFFIX`
140     echo "# $IPP_FMRI" >$file
141     service_is_enabled ${IPP_FMRI}
142     if [ $? -eq 0 ]; then
143         #
144         # If Listen directives are used, it's possible to listen on
145         # more than one ports. Process the Port directives only when Lis
146         # directives don't exist.
147         #
148         ipp_ports=`grep '^[ \t]*[^\# ]*Listen' ${IPP_CONF} | awk '{print

150         if [ -z "$ipp_ports" ]; then
151             ipp_ports=`grep '^[ \t]*[^\# ]*Port' ${IPP_CONF} | \
152                 awk '{print $2}' | tail -1`
153         fi

155         for port in $ipp_ports; do
156             generate_rules $FMRI $policy "tcp" $ip $port $file
157         done
158     fi

160     ;;

162 *)
163     echo "Usage: $0 { start | stop }"
164     exit 1
165     ;;
166 esac
167 exit $SMF_EXIT_OK
```

new/usr/src/lib/print/libipp-listener/common/ipp-listener.c

1

```
*****
14616 Wed May 7 18:40:48 2014
new/usr/src/lib/print/libipp-listener/common/ipp-listener.c
2837 - remove print/lp* from gate and use CUPS from userland
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21
22 /*
23  * Copyright 2008 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25  *
26  */
27
28 /* $Id: ipp-listener.c 146 2006-03-24 00:26:54Z njacobs $ */
29
30 #pragma ident "%Z%M% %I% %E% SMI"
31
32 #include <stdio.h>
33 #include <stdlib.h>
34 #include <string.h>
35 #include <netinet/in.h>
36 #include <assert.h>
37 #include <errno.h>
38 #include <syslog.h>
39 #include <sys/types.h>
40 #include <sys/stat.h>
41 #include <fcntl.h>
42 #include <unistd.h>
43 #include <sys/systeminfo.h>
44
45 #include <papi.h>
46 #include <ipp-listener.h>
47 #include <uri.h>
48
49 typedef papi_status_t (ipp_handler_t)(papi_service_t svc,
50                                     papi_attribute_t **request,
51                                     papi_attribute_t ***response,
52                                     ipp_reader_t iread, void *fd);
53
54 /*
55  * protocol request handlers are inserted below. The handler must be
56  * declared extern immediately below this comment and then an entry
57  * must be inserted in the "handlers" table a little further down.
58  */
59 extern ipp_handler_t ipp_print_job;
60 extern ipp_handler_t ipp_validate_job;
61 extern ipp_handler_t ipp_create_job;
```

new/usr/src/lib/print/libipp-listener/common/ipp-listener.c

2

```
62 extern ipp_handler_t ipp_get_printer_attributes;
63 extern ipp_handler_t ipp_get_jobs;
64 extern ipp_handler_t ipp_pause_printer;
65 extern ipp_handler_t ipp_resume_printer;
66 extern ipp_handler_t ipp_disable_printer;
67 extern ipp_handler_t ipp_enable_printer;
68 extern ipp_handler_t ipp_purge_jobs;
69 extern ipp_handler_t ipp_send_document;
70 extern ipp_handler_t ipp_cancel_job;
71 extern ipp_handler_t ipp_get_job_attributes;
72 extern ipp_handler_t ipp_release_job;
73 extern ipp_handler_t ipp_hold_job;
74 extern ipp_handler_t ipp_restart_job;
75 extern ipp_handler_t ipp_set_job_attributes;
76 extern ipp_handler_t ipp_set_printer_attributes;
77 extern ipp_handler_t cups_get_default;
78 extern ipp_handler_t cups_get_printers;
79 extern ipp_handler_t cups_get_classes;
80 extern ipp_handler_t cups_accept_jobs;
81 extern ipp_handler_t cups_reject_jobs;
82 extern ipp_handler_t cups_move_job;
83
84 /* ARGSUSED0 */
85 static papi_status_t
86 default_handler(papi_service_t svc, papi_attribute_t **request,
87                papi_attribute_t ***response, ipp_reader_t iread, void *fd)
88 {
89     int result = (int)PAPI_INTERNAL_ERROR;
90
91     if (response != NULL)
92         (void) papiAttributeListGetInteger(*response, NULL,
93                                           "status-code", &result);
94
95     return ((papi_status_t)result);
96 }
97
98 _unchanged_portion_omitted_
99
100 static int
101 ipp_operation_name_to_index(const char *name)
102 ipp_operation_name_to_index(char *name)
103 {
104     int i;
105
106     for (i = 0; handlers[i].name != NULL; i++)
107         if (strcasecmp(name, handlers[i].name) == 0)
108             return (i);
109
110     return (-1);
111 }
112
113 _unchanged_portion_omitted_
114
115 static char
116 type_to_boolean(const char *type)
117 type_to_boolean(char *type)
118 {
119     char result = PAPI_FALSE;
120
121     if ((strcasecmp(type, "true") == 0) ||
122         (strcasecmp(type, "yes") == 0) ||
123         (strcasecmp(type, "on") == 0) ||
124         (strcasecmp(type, "enable") == 0))
125         result = PAPI_TRUE;
126
127     return (result);
128 }
129
130 _unchanged_portion_omitted_
```

```
276 papi_status_t
277 ipp_configure_operation(papi_attribute_t ***list, const char *operation, const c
279 ipp_configure_operation(papi_attribute_t ***list, char *operation, char *type)
278 {
279     papi_status_t result = PAPI_OPERATION_NOT_SUPPORTED;
280     char boolean = PAPI_FALSE;
281
282     if ((list == NULL) || (operation == NULL) || (type == NULL))
283         return (PAPI_BAD_ARGUMENT);
284
285     boolean = type_to_boolean(type);
286
287     if (strcasecmp(operation, "all") == 0) {
288         result = ipp_configure_all_operations(list, boolean);
289     } else if (strcasecmp(operation, "required") == 0) {
290         result = ipp_configure_required_operations(list, boolean);
291     } else if (ipp_operation_name_to_index(operation) != -1) {
292         result = papiAttributeListAddBoolean(list, PAPI_ATTR_REPLACE,
293                                             operation, boolean);
294     }
295
296     return (result);
297 }
unchanged_portion_omitted
```

new/usr/src/lib/print/libipp-listener/common/ipp-listener.h

1

```
*****
2293 Wed May 7 18:40:49 2014
new/usr/src/lib/print/libipp-listener/common/ipp-listener.h
2837 - remove print/lp* from gate and use CUPS from userland
*****
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 2006 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25  *
26  */

28 #ifndef _IPP_LISTENER_H
29 #define _IPP_LISTENER_H

31 /* $Id: ipp-listener.h 146 2006-03-24 00:26:54Z njacobs $ */

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

33 #ifdef __cplusplus
34 extern "C" {
35 #endif

37 #include <ipp.h>

39 /* exported functions */
40 extern papi_status_t ipp_configure_operation(papi_attribute_t ***list,
41                                             const char *operation,
42                                             const char *type);
43                                             char *operation, char *type);
44 extern papi_status_t ipp_process_request(papi_attribute_t **request,
45                                         papi_attribute_t ***response,
46                                         ipp_reader_t iread, void *fd);

47 /* shared internal functions */
48 extern char *ipp_svc_status_mess(papi_service_t svc, papi_status_t status);
49 extern char *destination_from_printer_uri(char *);
50 extern void get_printer_id(papi_attribute_t **attributes, char **printer,
51                             int *id);
52 extern void ipp_operations_supported(papi_attribute_t ***list,
53                                     papi_attribute_t **request);
54 extern void get_string_list(papi_attribute_t **attributes, char *name,
55                             char ***values);
56 extern void add_default_attributes(papi_attribute_t ***attributes);
57 extern void papi_to_ipp_printer_group(papi_attribute_t ***response,
58                                       papi_attribute_t **request, int flags,
```

new/usr/src/lib/print/libipp-listener/common/ipp-listener.h

2

```
59         papi_printer_t p);
60 extern void papi_to_ipp_job_group(papi_attribute_t ***response,
61                                   papi_attribute_t **request, int flags, papi_job_t j);
62 extern void message_response(papi_attribute_t **request,
63                               papi_attribute_t **response);

65 #ifdef __cplusplus
66 }
_____unchanged_portion_omitted_____
```

```

new/usr/src/lib/print/libpapi-common/common/attribute.c 1
*****
25605 Wed May 7 18:40:49 2014
new/usr/src/lib/print/libpapi-common/common/attribute.c
2837 - remove print/lp* from gate and use CUPS from userland
*****
_____unchanged_portion_omitted_____

205 static papi_attribute_t *
206 papiAttributeAlloc(const char *name, papi_attribute_value_type_t type)
206 papiAttributeAlloc(char *name, papi_attribute_value_type_t type)
207 {
208     papi_attribute_t *result = NULL;

210     if ((result = calloc(1, sizeof (*result))) != NULL) {
211         result->name = strdup(name);
212         result->type = type;
213     }

215     return (result);
216 }
_____unchanged_portion_omitted_____

239 papi_status_t
240 papiAttributeListAddValue(papi_attribute_t ***list, int flgs,
241     const char *name, papi_attribute_value_type_t type,
242     char *name, papi_attribute_value_type_t type,
243     papi_attribute_value_t *value)
243 {
244     papi_status_t result;
245     int flags = flgs;
246     papi_attribute_t *attribute = NULL;
247     papi_attribute_value_t **values = NULL;

249     if ((list == NULL) || (name == NULL))
250         return (PAPI_BAD_ARGUMENT);

252     if ((type == PAPI_RANGE) && (value != NULL) &&
253         (value->range.lower > value->range.upper))
254         return (PAPI_BAD_ARGUMENT); /* RANGE must have min <= max */

256     if (flags == 0) /* if it wasn't set, set a default behaviour */
257         flags = PAPI_ATTR_APPEND;

259     /* look for an existing one */
260     attribute = papiAttributeListFind(*list, name);

262     if (((flags & PAPI_ATTR_EXCL) != 0) && (attribute != NULL))
263         return (PAPI_CONFLICT); /* EXISTS */

265     if (((flags & PAPI_ATTR_REPLACE) == 0) && (attribute != NULL) &&
266         (attribute->type != type))
267         return (PAPI_CONFLICT); /* TYPE CONFLICT */

269     /* if we don't have one, create it and add it to the list */
270     if ((attribute == NULL) &&
271         ((attribute = papiAttributeAlloc(name, type)) != NULL))
272         list_append(list, attribute);

274     /* if we don't have one by now, it's most likely an alloc fail */
275     if (attribute == NULL)
276         return (PAPI_TEMPORARY_ERROR);

278     /*
279     * if we are replacing, clear any existing values, but don't free
280     * until after we have replaced the values, in case we are replacing
281     * a collection with a relocated version of the original collection.

```

```

new/usr/src/lib/print/libpapi-common/common/attribute.c 2
282     */
283     if (((flags & PAPI_ATTR_REPLACE) != 0) && (attribute->values != NULL)) {
284         values = attribute->values;
285         attribute->values = NULL;
286     }

288     attribute->type = type;

290     result = papiAttributeListAppendValue(&attribute->values, type, value);

292     /* free old values if we replaced them */
293     if (values != NULL)
294         papiAttributeValuesFree(type, values);

296     return (result);
297 }
_____unchanged_portion_omitted_____

319 papi_status_t
320 papiAttributeListAddBoolean(papi_attribute_t ***list, int flags,
321     const char *name, char boolean)
322     char *name, char boolean)
322 {
323     papi_attribute_value_t v;

325     v.boolean = boolean;
326     return (papiAttributeListAddValue(list, flags, name, PAPI_BOOLEAN, &v));
327 }
_____unchanged_portion_omitted_____

404 papi_attribute_t *
405 papiAttributeListFind(papi_attribute_t **list, const char *name)
405 papiAttributeListFind(papi_attribute_t **list, char *name)
406 {
407     int i;
408     if ((list == NULL) || (name == NULL))
409         return (NULL);

411     for (i = 0; list[i] != NULL; i++)
412         if (strcasecmp(list[i]->name, name) == 0)
413             return ((papi_attribute_t *)list[i]);

415     return (NULL);
416 }
_____unchanged_portion_omitted_____

```

```

*****
15706 Wed May 7 18:40:49 2014
new/usr/src/lib/print/libpapi-common/common/papi.h
2837 - remove print/lp* from gate and use CUPS from userland
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */

22 /*
23  * Copyright 2008 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25  *
26  */

28 #ifndef _PAPI_H
29 #define _PAPI_H

31 /* $Id: papi.h 161 2006-05-03 04:32:59Z njacobs $ */

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

33 #include <sys/types.h>
34 #include <time.h>
35 #include <stdio.h>

37 #ifdef __cplusplus
38 extern "C" {
39 #endif

41 /*
42  * Types
43  */

45 /*      service related types      */
46 typedef void *papi_service_t;
47 typedef void *papi_printer_t;
48 typedef void *papi_job_t;
49 typedef void *papi_stream_t;

51 typedef enum {
52     PAPI_ENCRYPT_IF_REQUESTED, /* Encrypt if requested (TLS upgrade) */
53     PAPI_ENCRYPT_NEVER,      /* Never encrypt */
54     PAPI_ENCRYPT_REQUIRED,   /* Encryption required (TLS upgrade) */
55     PAPI_ENCRYPT_ALWAYS      /* Always encrypt (SSL) */
56 } papi_encryption_t;
      unchanged portion omitted
223 /*

```

```

224 * Functions
225 */

227 /*      Service related      */
228 extern papi_status_t papiServiceCreate(papi_service_t *handle,
229                                       char *service_name, char *user_name,
230                                       char *password,
231                                       int (*authCB)(papi_service_t svc,
232                                                     void *app_data),
233                                       papi_encryption_t encryption,
234                                       void *app_data);
235 extern void papiServiceDestroy(papi_service_t handle);
236 extern papi_status_t papiServiceSetUserName(papi_service_t handle,
237                                             char *user_name);
238 extern papi_status_t papiServiceSetPassword(papi_service_t handle,
239                                             char *password);
240 extern papi_status_t papiServiceSetEncryption(papi_service_t handle,
241                                               papi_encryption_t encryption);
242 extern papi_status_t papiServiceSetAuthCB(papi_service_t handle,
243                                           int (*authCB)(papi_service_t s,
244                                                         void *app_data));
245 extern papi_status_t papiServiceSetAppData(papi_service_t handle,
246                                             void *app_data);
247 extern char *papiServiceGetServiceName(papi_service_t handle);
248 extern char *papiServiceGetUserName(papi_service_t handle);
249 extern char *papiServiceGetPassword(papi_service_t handle);
250 extern papi_encryption_t papiServiceGetEncryption(papi_service_t handle);
251 extern void *papiServiceGetAppData(papi_service_t handle);
252 extern papi_attribute_t **papiServiceGetAttributeList(papi_service_t handle);
253 extern char *papiServiceGetStatusMessage(papi_service_t handle);

255 /*      Attribute related      */
256 extern papi_status_t papiAttributeListAddValue(papi_attribute_t ***attrs,
257                                                int flags, const char *name,
258                                                int flags, char *name,
259                                                papi_attribute_value_type_t type,
260                                                papi_attribute_value_t *value);
261 extern papi_status_t papiAttributeListAddString(papi_attribute_t ***attrs,
262                                                 int flags, char *name, char *string);
263 extern papi_status_t papiAttributeListAddInteger(papi_attribute_t ***attrs,
264                                                  int flags, char *name, int integer);
265 extern papi_status_t papiAttributeListAddBoolean(papi_attribute_t ***attrs,
266                                                  int flags, const char *name, char boolean);
267 extern papi_status_t papiAttributeListAddRange(papi_attribute_t ***attrs,
268                                                int flags, char *name,
269                                                int lower, int upper);
270 extern papi_status_t papiAttributeListAddResolution(papi_attribute_t ***attrs,
271                                                     int flags, char *name,
272                                                     int xres, int yres,
273                                                     papi_resolution_unit_t units);
274 extern papi_status_t papiAttributeListAddDatetime(papi_attribute_t ***attrs,
275                                                   int flags, char *name, time_t datetime);
276 extern papi_status_t papiAttributeListAddCollection(papi_attribute_t ***attrs,
277                                                    int flags, char *name,
278                                                    papi_attribute_t **collection);
279 extern papi_status_t papiAttributeListAddMetadata(papi_attribute_t ***attrs,
280                                                  int flags, char *name,
281                                                  papi_metadata_t metadata);
282 extern papi_status_t papiAttributeListDelete(papi_attribute_t ***attributes,
283                                             char *name);
284 extern papi_status_t papiAttributeListGetValue(papi_attribute_t **list,
285                                                void **iterator, char *name,
286                                                papi_attribute_value_type_t type,
287                                                papi_attribute_value_t **value);
288 extern papi_status_t papiAttributeListGetString(papi_attribute_t **list,

```

```

288         void **iterator, char *name,
289         char **vptr);
290 extern papi_status_t papiAttributeListGetInteger(papi_attribute_t **list,
291         void **iterator, char *name, int *vptr);
292 extern papi_status_t papiAttributeListGetBoolean(papi_attribute_t **list,
293         void **iterator, char *name,
294         char **vptr);
295 extern papi_status_t papiAttributeListGetRange(papi_attribute_t **list,
296         void **iterator, char *name,
297         int *min, int *max);
298 extern papi_status_t papiAttributeListGetResolution(papi_attribute_t **list,
299         void **iterator, char *name,
300         int *x, int *y,
301         papi_resolution_unit_t *units);
302 extern papi_status_t papiAttributeListGetDatetime(papi_attribute_t **list,
303         void **iterator, char *name,
304         time_t *dt);
305 extern papi_status_t papiAttributeListGetCollection(papi_attribute_t **list,
306         void **iterator, char *name,
307         papi_attribute_t ***collection);
308 extern papi_status_t papiAttributeListGetMetadata(papi_attribute_t **list,
309         void **iterator, char *name,
310         papi_metadata_t *vptr);
311 extern papi_attribute_t *papiAttributeListFind(papi_attribute_t **list,
312         const char *name);
313 extern papi_attribute_t *papiAttributeListGetNext(papi_attribute_t **list,
314         void **iterator);
315 extern void papiAttributeListFree(papi_attribute_t **attributes);

317 extern papi_status_t papiAttributeListFromString(papi_attribute_t **attrs,
318         int flags, char *string);
319 extern papi_status_t papiAttributeListToString(papi_attribute_t **attrs,
320         char *delim,
321         char *buffer, size_t buflen);
322 extern void papiAttributeListPrint(FILE *fp, papi_attribute_t **list,
323         char *prefix_fmt, ...);

325 /*      Printer related      */
326 extern papi_status_t papiPrintersList(papi_service_t handle,
327         char **requested_attrs,
328         papi_filter_t *filter,
329         papi_printer_t **printers);
330 extern papi_status_t papiPrinterQuery(papi_service_t handle, char *name,
331         char **requested_attrs,
332         papi_attribute_t **job_attributes,
333         papi_printer_t *printer);
334 extern papi_status_t papiPrinterAdd(papi_service_t handle, char *name,
335         papi_attribute_t **attributes,
336         papi_printer_t *printer);
337 extern papi_status_t papiPrinterModify(papi_service_t handle, char *name,
338         papi_attribute_t **attributes,
339         papi_printer_t *printer);
340 extern papi_status_t papiPrinterRemove(papi_service_t handle, char *name);
341 extern papi_status_t papiPrinterDisable(papi_service_t handle, char *name,
342         char *message);
343 extern papi_status_t papiPrinterEnable(papi_service_t handle, char *name);
344 extern papi_status_t papiPrinterPause(papi_service_t handle, char *name,
345         char *message);
346 extern papi_status_t papiPrinterResume(papi_service_t handle, char *name);
347 extern papi_status_t papiPrinterPurgeJobs(papi_service_t handle,
348         char *name, papi_job_t **jobs);
349 extern papi_status_t papiPrinterListJobs(papi_service_t handle,
350         char *name, char **requested_attrs,
351         int type_mask, int max_num_jobs,
352         papi_job_t **jobs);

```

```

353 extern papi_attribute_t **papiPrinterGetAttributeList(papi_printer_t printer);
354 extern void papiPrinterFree(papi_printer_t printer);
355 extern void papiPrinterListFree(papi_printer_t *printers);

357 /*      Job related      */
358 extern papi_status_t papiJobSubmit(papi_service_t handle, char *printer,
359         papi_attribute_t **job_attributes,
360         papi_job_ticket_t *job_ticket,
361         char **files, papi_job_t *job);
362 extern papi_status_t papiJobSubmitByReference(papi_service_t handle,
363         char *printer,
364         papi_attribute_t **job_attributes,
365         papi_job_ticket_t *job_ticket,
366         char **files, papi_job_t *job);
367 extern papi_status_t papiJobValidate(papi_service_t handle, char *printer,
368         papi_attribute_t **job_attributes,
369         papi_job_ticket_t *job_ticket,
370         char **files, papi_job_t *job);
371 extern papi_status_t papiJobStreamOpen(papi_service_t handle,
372         char *printer,
373         papi_attribute_t **job_attributes,
374         papi_job_ticket_t *job_ticket,
375         papi_stream_t *stream);
376 extern papi_status_t papiJobStreamWrite(papi_service_t handle,
377         papi_stream_t stream,
378         void *buffer, size_t buflen);
379 extern papi_status_t papiJobStreamClose(papi_service_t handle,
380         papi_stream_t stream,
381         papi_job_t *job);
382 extern papi_status_t papiJobQuery(papi_service_t handle, char *printer,
383         int32_t job_id, char **requested_attrs,
384         papi_job_t *job);
385 extern papi_status_t papiJobModify(papi_service_t handle, char *printer,
386         int32_t job_id,
387         papi_attribute_t **attributes,
388         papi_job_t *job);
389 extern papi_status_t papiJobMove(papi_service_t handle, char *printer,
390         int32_t job_id, char *destination);
391 extern papi_status_t papiJobCancel(papi_service_t handle, char *printer,
392         int32_t job_id);
393 extern papi_status_t papiJobHold(papi_service_t handle, char *printer,
394         int32_t job_id);
395 extern papi_status_t papiJobRelease(papi_service_t handle, char *printer,
396         int32_t job_id);
397 extern papi_status_t papiJobRestart(papi_service_t handle, char *printer,
398         int32_t job_id);
399 extern papi_status_t papiJobPromote(papi_service_t handle, char *printer,
400         int32_t job_id);
401 extern papi_attribute_t **papiJobGetAttributeList(papi_job_t printer);
402 extern char *papiJobGetPrinterName(papi_job_t printer);
403 extern int32_t papiJobGetId(papi_job_t printer);
404 extern papi_job_ticket_t *papiJobGetJobTicket(papi_job_t printer);
405 extern void papiJobFree(papi_job_t job);
406 extern void papiJobListFree(papi_job_t *jobs);

408 #ifdef SOLARIS_PRIVATE_POST_0_9
409 /*
410 * These have been added to support IPP create-job/send-document with PAPI v0.9
411 * in an IPP listener using PAPI as it's spooler interface. A future version
412 * of the API is expected to support this type of functionality
413 */
414 extern papi_status_t papiJobCreate(papi_service_t handle, char *printer,
415         papi_attribute_t **job_attributes,
416         papi_job_ticket_t *job_ticket,
417         papi_job_t *job);
418 extern papi_status_t papiJobStreamAdd(papi_service_t handle, char *printer,

```



```
419             int32_t id, papi_stream_t *stream);
420 extern papi_status_t papiJobCommit(papi_service_t handle, char *printer,
421             int32_t id);
422 extern papi_status_t papiServiceSetPeer(papi_service_t handle, int peerfd);
423 #endif /* SOLARIS_PRIVATE_POST_0_9 */

425 extern char *papiStatusString(papi_status_t status);

427 /*
428 * Internal functions that aren't in the API, but are shared across
429 * protocol support implementations(psms) and the tightly bound
430 * listener library. Do not use these in your applications.
431 */
432 extern void list_append();
433 extern void list_concatenate();
434 extern void list_remove();
435 extern void copy_attributes(papi_attribute_t ***result,
436             papi_attribute_t **list);
437 extern void split_and_copy_attributes(char **list,
438             papi_attribute_t **attributes,
439             papi_attribute_t ***in,
440             papi_attribute_t ***out);

442 extern papi_attribute_t **getprinterbyname(char *name, char *ns);

444 extern int is_localhost(char *hostname);

446 #ifdef __cplusplus
447 }
448 unchanged_portion_omitted
```

new/usr/src/lib/print/mod\_ipp/Makefile

1

\*\*\*\*\*

2580 Wed May 7 18:40:49 2014

new/usr/src/lib/print/mod\_ipp/Makefile

2837 - remove print/lp\* from gate and use CUPS from userland

\*\*\*\*\*

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

26 LIBRARY =          mod_ipp.a
27 VERS =
28 OBJECTS = mod_ipp.o

30 include ../../Makefile.lib
31 include ../../Makefile.rootfs

33 APACHEMODDIR = $(ROOT)/usr/apache2/2.2/libexec
34 APACHECONFDIR = $(ROOT)/etc/apache2/2.2
33 APACHEMODDIR = $(ROOT)/usr/apache/libexec
34 APACHECONFDIR = $(ROOT)/etc/apache
35 LISTENERDIR = $(ROOT)/var/lp/ipp-listener

37 ROOTDIRS =         $(ROOT)/usr/apache2/2.2 $(APACHEMODDIR) $(APACHECONFDIR) \
37 ROOTDIRS =         $(ROOT)/usr/apache $(APACHEMODDIR) $(APACHECONFDIR) \
38                   $(ROOT)/var/lp $(LISTENERDIR)

40 $(ROOT)/var/lp:=   DIRMODE = 775
41 $(ROOT)/var/lp:=   FILEMODE = 775

43 LIBS =             $(DYNLIB)

45 SRCS = $(OBJECTS:%.o = %.c)

47 APR_INC.cmd = apr-1-config --includes
48 APU_INC.cmd = apu-1-config --includes
49 CFLAGS +=          $(CCVERBOSE)
50 #endif /* ! codereview */

47 CFLAGS +=         $(CCVERBOSE)
52 CPPFLAGS +=       -I../libipp-listener/common
53 CPPFLAGS +=       -I../libipp-core/common
54 CPPFLAGS +=       -Iusr/apache2/2.2/include
55 CPPFLAGS +=       $(APR_INC.cmd:sh)
56 CPPFLAGS +=       $(APU_INC.cmd:sh)
50 CPPFLAGS +=       -Iusr/apache/include
```

new/usr/src/lib/print/mod\_ipp/Makefile

2

```
57 CPPFLAGS +=      -DEAPI
58 ZDEFS =           $(ZNODEFS)

60 MAPFILES =        mapfile

62 LDLIBS +=         -lipp-listener -lipp-core -lpapi -lc

64 #               SMF manifest
65 MANIFEST=         ipp-listener.xml
66 ROOTMANIFESTDIR= $(ROOT)/lib/svc/manifest/application/print
67 ROOTMANIFEST=     $(MANIFEST:%=$(ROOTMANIFESTDIR)/%)
68 $(ROOTMANIFEST)  := FILEMODE= 444

70 #               Apache module
71 $(APACHEMODDIR)/$(LIBLINKS): $(ROOTDIRS)

73 #               Apache config
74 APACHECONFFILE=   $(APACHECONFDIR)/httpd-standalone-ipp.conf
75 $(APACHECONFFILE) := FILEMODE= 644
76 LISTENERFILE=    $(LISTENERDIR)/index.html
77 $(LISTENERFILE)  := FILEMODE= 444

79 $(ROOT)/var/lp:=   FILEMODE = 0775

81 $(APACHEMODDIR)/$(LIBLINKS):= FILEMODE = 0555

83 $(ROOTMANIFESTDIR)/% $(APACHEMODDIR)/% $(APACHECONFDIR)/% $(LISTENERDIR)/%:
84     $(INS.file)

86 $(ROOTDIRS):
87     $(INS.dir)

89 .KEEP_STATE:

91 all:             $(LIBS)

93 install:         all $(APACHEMODDIR)/$(LIBLINKS) $(APACHECONFFILE) \
94                 $(LISTENERFILE) $(ROOTMANIFEST)

96 install_h:

98 lint:

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

new/usr/src/lib/print/mod\_ipp/httpd-standalone-ipp.conf

1

```
*****
11280 Wed May 7 18:40:50 2014
new/usr/src/lib/print/mod_ipp/httpd-standalone-ipp.conf
2837 - remove print/lp* from gate and use CUPS from userland
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2006 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #
27 #
28 # "$Id: httpd-standalone-ipp.conf,v 1.4 2006/03/24 00:26:54 njacobs Exp $"
29 #
30 #
31 # ident "%Z%M% %I% %E% SMI"
32 #
33 ##
34 ## httpd-standalone-ipp.conf -- Apache HTTP server configuration file for
35 ## an Internet Print Protocol (IPP) listener
36 ##
37 #
38 #
39 # Based upon the NCSA server configuration files originally by Rob McCool.
40 #
41 # This is the main Apache server configuration file. It contains the
42 # configuration directives that give the server its instructions.
43 # See <URL:http://www.apache.org/docs/> for detailed information about
44 # the directives. mod_ipp specific directives are described in the
45 # mod_ipp(4) man page.
46 #
47 #
48 ### Section 1: Global Environment
49 #
50 # The directives in this section affect the overall operation of Apache,
51 # such as the number of concurrent requests it can handle or where it
52 # can find its configuration files.
53 #
54 #
55 #
56 # ServerType is either inetd, or standalone. Inetd mode is only supported on
57 # Unix platforms.
58 #
59 ServerType standalone
60 #
61 #
```

new/usr/src/lib/print/mod\_ipp/httpd-standalone-ipp.conf

2

```
56 # ServerRoot: The top of the directory tree under which the server's
57 # configuration, error, and log files are kept.
58 #
59 # NOTE! If you intend to place this on an NFS (or otherwise network)
60 # mounted filesystem then please read the LockFile documentation
61 # (available at <URL:http://www.apache.org/docs/mod/core.html#lockfile>);
62 # you will save yourself a lot of trouble.
63 #
64 ServerRoot "/usr/apache2/2.2"
65 ServerRoot "/usr/apache"
66 #
67 # The LockFile directive sets the path to the lockfile used when Apache
68 # is compiled with either USE_FCNTL_SERIALIZED_ACCEPT or
69 # USE_FLOCK_SERIALIZED_ACCEPT. This directive should normally be left at
70 # its default value. The main reason for changing it is if the logs
71 # directory is NFS mounted, since the lockfile MUST BE STORED ON A LOCAL
72 # DISK. The PID of the main server process is automatically appended to
73 # the filename.
74 #
75 #LockFile /var/run/httpd.lock
76 LockFile /var/run/httpd-standalone-ipp.lock
77 #
78 #
79 # PidFile: The file in which the server should record its process
80 # identification number when it starts.
81 #
82 PidFile /var/run/httpd-standalone-ipp.pid
83 #
84 #
85 # ScoreBoardFile: File used to store internal server process information.
86 # Not all architectures require this. But if yours does (you'll know because
87 # this file will be created when you run Apache) then you *must* ensure that
88 # no two invocations of Apache share the same scoreboard file.
89 #
90 ScoreBoardFile /var/run/httpd-standalone-ipp.scoreboard
91 #
92 #
93 # In the standard configuration, the server will process httpd.conf (this
94 # file, specified by the -f command line option), srm.conf, and access.conf
95 # in that order. The latter two files are now distributed empty, as it is
96 # recommended that all directives be kept in a single file for simplicity.
97 # The commented-out values below are the built-in defaults. You can have the
98 # server ignore these files altogether by using "/dev/null" (for Unix) or
99 # "nul" (for Win32) for the arguments to the directives.
100 #
101 #ResourceConfig conf/srm.conf
102 #AccessConfig conf/access.conf
103 #
104 #
105 # Timeout: The number of seconds before receives and sends time out.
106 #
107 Timeout 300
108 #
109 #
110 # KeepAlive: Whether or not to allow persistent connections (more than
111 # one request per connection). Set to "Off" to deactivate.
112 #
113 KeepAlive On
114 #
115 #
116 # MaxKeepAliveRequests: The maximum number of requests to allow
117 # during a persistent connection. Set to 0 to allow an unlimited amount.
118 # We recommend you leave this number high, for maximum performance.
119 #
120 MaxKeepAliveRequests 100
```

```

122 #
123 # KeepAliveTimeout: Number of seconds to wait for the next request from the
124 # same client on the same connection.
125 #
126 KeepAliveTimeout 15

128 #
129 # Server-pool size regulation. Rather than making you guess how many
130 # server processes you need, Apache dynamically adapts to the load it
131 # sees --- that is, it tries to maintain enough server processes to
132 # handle the current load, plus a few spare servers to handle transient
133 # load spikes (e.g., multiple simultaneous requests from a single
134 # Netscape browser).
135 #
136 # It does this by periodically checking how many servers are waiting
137 # for a request. If there are fewer than MinSpareServers, it creates
138 # a new spare. If there are more than MaxSpareServers, some of the
139 # spares die off. The default values are probably OK for most sites.
140 #
141 MinSpareServers 1
142 MaxSpareServers 2

144 #
145 # Number of servers to start initially --- should be a reasonable ballpark
146 # figure.
147 #
148 StartServers 1

150 #
151 # Limit on total number of servers running, i.e., limit on the number
152 # of clients who can simultaneously connect --- if this limit is ever
153 # reached, clients will be LOCKED OUT, so it should NOT BE SET TOO LOW.
154 # It is intended mainly as a brake to keep a runaway server from taking
155 # the system with it as it spirals down...
156 #
157 MaxClients 150

159 #
160 # MaxRequestsPerChild: the number of requests each child process is
161 # allowed to process before the child dies. The child will exit so
162 # as to avoid problems after prolonged use when Apache (and maybe the
163 # libraries it uses) leak memory or other resources. On most systems, this
164 # isn't really needed, but a few (such as Solaris) do have notable leaks
165 # in the libraries. For these platforms, set to something like 10000
166 # or so; a setting of 0 means unlimited.
167 #
168 # NOTE: This value does not include keepalive requests after the initial
169 # request per connection. For example, if a child process handles
170 # an initial request and 10 subsequent "keptalive" requests, it
171 # would only count as 1 request towards this limit.
172 #
173 MaxRequestsPerChild 10

175 #
176 # Dynamic Shared Object (DSO) Support
177 #
178 # To be able to use the functionality of a module which was built as a DSO you
179 # have to place corresponding 'LoadModule' lines at this location so the
180 # directives contained in it are actually available _before_ they are used.
181 # Please read the file http://httpd.apache.org/docs/dso.html for more
182 # details about the DSO mechanism and run 'httpd -l' for the list of already
183 # built-in (statically linked and thus always available) modules in your httpd
184 # binary.
185 #
186 # Note: The order in which modules are loaded is important. Don't change

```

```

187 # the order below without expert advice.
188 #
189 LoadModule access_module libexec/mod_access.so
190 LoadModule alias_module libexec/mod_alias.so
191 LoadModule auth_basic_module libexec/mod_auth_basic.so
192 LoadModule authn_file_module libexec/mod_authn_file.so
193 LoadModule authz_host_module libexec/mod_authz_host.so
194 LoadModule authz_user_module libexec/mod_authz_user.so
195 LoadModule auth_module libexec/mod_auth.so
196 LoadModule mime_module libexec/mod_mime.so
197 LoadModule mime_magic_module libexec/mod_mime_magic.so
198 LoadModule ipp_module libexec/mod_ipp.so

202 # Reconstruction of the complete module list from all available modules
203 # (static and shared ones) to achieve correct module execution order.
204 # [WHENEVER YOU CHANGE THE LOADMODULE SECTION ABOVE UPDATE THIS, TOO]
205 ClearModuleList
206 AddModule mod_access.c
207 AddModule mod_alias.c
208 AddModule mod_auth.c
209 AddModule mod_mime.c
210 AddModule mod_mime_magic.c
211 AddModule mod_ipp.c
212 AddModule mod_so.c

198 ### Section 2: 'Main' server configuration
199 #
200 # The directives in this section set up the values used by the 'main'
201 # server, which responds to any requests that aren't handled by a
202 # <VirtualHost> definition. These values also provide defaults for
203 # any <VirtualHost> containers you may define later in the file.
204 #
205 # All of these directives may appear inside <VirtualHost> containers,
206 # in which case these default settings will be overridden for the
207 # virtual host being defined.
208 #

210 #
211 # If your ServerType directive (set earlier in the 'Global Environment'
212 # section) is set to "inetd", the next few directives don't have any
213 # effect since their settings are defined by the inetd configuration.
214 # Skip ahead to the ServerAdmin directive.
215 #

217 #
218 # Port: The port to which the standalone server listens. For
219 # ports < 1023, you will need httpd to be run as root initially.
220 #
221 Listen 631
222 #
223 #
224 # If you wish httpd to run as a different user or group, you must run
225 # httpd as root initially and it will switch.
226 #
227 # User/Group: The name (or #number) of the user/group to run httpd as.
228 # . On SCO (ODT 3) use "User nouser" and "Group nogroup".
229 # . On HPUX you may not be able to use shared memory as nobody, and the
230 # suggested workaround is to create a user www and use that user.
231 # NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
232 # when the value of (unsigned)Group is above 60000;
233 # don't use Group nobody on these systems!
234 #
235 User lp
236 Group lp

```

```

238 #
239 # ServerAdmin: Your address, where problems with the server should be
240 # e-mailed. This address appears on some server-generated pages, such
241 # as error documents.
242 #
243 ServerAdmin lp@localhost

245 #
246 # ServerName allows you to set a host name which is sent back to clients for
247 # your server if it's different than the one the program would get (i.e., use
248 # "www" instead of the host's real name).
249 #
250 # Note: You cannot just invent host names and hope they work. The name you
251 # define here must be a valid DNS name for your host. If you don't understand
252 # this, ask your network administrator.
253 # If your host doesn't have a registered DNS name, enter its IP address here.
254 # You will have to access it by its address (e.g., http://123.45.67.89/)
255 # anyway, and this will make redirections work in a sensible way.
256 #
257 # 127.0.0.1 is the TCP/IP local loop-back address, often named localhost. Your
258 # machine always knows itself by this address. If you use Apache strictly for
259 # local testing and development, you may use 127.0.0.1 as the server name.
260 #
261 #Servername printserver.some_company.com

263 DefaultType application/ipp

265 ErrorLog      /var/lp/logs/ipp-errors
266 LogLevel     warn

268 DocumentRoot /var/lp/ipp-listener

270 # Allow passing PPD files from this service as well
271 Alias /etc/lp/ppd/ /etc/lp/ppd/
272 <Directory /etc/lp/ppd>
273     SetHandler send-as-is
274     <LimitExcept GET>
275         Deny from all
276     </LimitExcept>
277 </Directory>

279 # mod_ipp specific configuration
280 <IfModule mod_ipp.c>

282     <Location />
283         # ipp-conformance automatic      # default
284         # ipp-default-user nobody
285         ipp-default-service lpsched
286         #
287         # By default, only turn on operations that are not
288         # likely to cause real problems when the user can't
289         # be trusted.
290         #
291         ipp-operation all off
292         ipp-operation print-job on
293         ipp-operation validate-job on
294         ipp-operation create-job on
295         ipp-operation get-jobs on
296         ipp-operation get-printer-attributes on
297         ipp-operation send-document on
298         ipp-operation cancel-job on
299         ipp-operation get-job-attributes on
300         ipp-operation cups-get-default on
301         ipp-operation cups-get-printers on
302         ipp-operation cups-get-classes on
303         ipp-operation cups-move-job on

```

```

305         # redirect non-IPP requests
306         ErrorDocument 404 /index.html
307     </Location>

309     <Location /admin>
310         # ipp-conformance automatic      # default
311         # ipp-default-user nobody
312         ipp-default-service lpsched

314         ipp-operation all on

316         AuthType Basic
317         AuthName "IPP Server"
318         AuthUserFile /etc/ipp-users
319         Require valid-user

321         # redirect non-IPP requests
322         ErrorDocument 404 /index.html
323     </Location>
324 </IfModule>

```

```

*****
2487 Wed May 7 18:40:50 2014
new/usr/src/lib/print/mod_ipp/ipp-listener.xml
2837 - remove print/lp* from gate and use CUPS from userland
*****
1 <?xml version="1.0"?>
2 <!--
3 CDDL HEADER START

5 The contents of this file are subject to the terms of the
6 Common Development and Distribution License (the "License").
7 You may not use this file except in compliance with the License.

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.

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]

20 CDDL HEADER END
21 -->
22 <!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
23 <!--
24 Copyright 2009 Sun Microsystems, Inc. All rights reserved.
25 Use is subject to license terms.
26 -->

28 <service_bundle type='manifest' name='SUNWipplr:ipp-listener'>

30 <service
31   name='application/print/ipp-listener'
32   type='service'
33   version='1'>

35   <create_default_instance enabled='false' />

37   <dependency name='print-service'
38     grouping='require_any'
39     restart_on='refresh'
40     type='service'>
41     <service_fmri value='svc:/application/print/server' />
42   </dependency>

44   <exec_method
45     type='method'
46     name='start'
47     exec='/usr/apache2/2.2/bin/httpd -f /etc/apache2/2.2/httpd-standalon
47     exec='/usr/apache/bin/httpd -f /etc/apache/httpd-standalone-ipp.conf
48     timeout_seconds='10' />

50   <exec_method
51     type='method'
52     name='stop'
53     exec='/bin/pkill -f httpd-standalone-ipp.conf'
54     timeout_seconds='5' />

56   <property_group name='general' type='framework'>
57     <!-- to start/stop IPP listening service-->
58     <propval name='action_authorization' type='astring'
59       value='solaris.print.admin' />
60     <propval name='value_authorization' type='astring'

```

```

61     value='solaris.print.admin' />
62   </property_group>

64   <property_group name='firewall_context' type='com.sun.fw_definition'>
65     <propval name='ipf_method' type='astring'
66       value='/lib/svc/method/print-svc ipfilter svc:/applicati
67   </property_group>

69   <stability value='Unstable' />

71   <template>
72     <common_name>
73       <loctext xml:lang='C'>
74         Internet Print Protocol Listening Service
75       </loctext>
76     </common_name>
77     <documentation>
78       <manpage title='mod_ipp' section='4'
79         manpath='/usr/share/man' />
80     </documentation>
81   </template>
82 </service>

84 </service_bundle>

```

new/usr/src/lib/print/mod\_ipp/mod\_ipp.c

1

```
*****
16057 Wed May 7 18:40:52 2014
new/usr/src/lib/print/mod_ipp/mod_ipp.c
2837 - remove print/lp* from gate and use CUPS from userland
*****
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 2006 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25  *
26  */
27 /*
28  * Copyright (c) 2013 Andrew Stormont. All rights reserved.
29  */
30 #endif /* ! codereview */

32 /* $Id: mod_ipp.c 149 2006-04-25 16:55:01Z njacobs $ */

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

34 /*
35  * Internet Printing Protocol (IPP) module for Apache.
36  */

38 #include "ap_config.h"

40 #include <stdio.h>
41 #include <time.h>
42 #include <sys/time.h>
43 #include <values.h>
44 #include <libintl.h>
45 #include <alloca.h>

47 #include "httpd.h"
48 #include "http_config.h"
49 #include "http_core.h"
50 #include "http_protocol.h"
51 #include "http_log.h"
52 #include "http_main.h"
53 #include "papi.h"
54 /* Try and guess the version of apache */
55 #ifdef APACHE_RELEASE
56 # define AP_SERVER_MAJORVERSION_NUMBER 1
57 # define AP_SERVER_MINORVERSION_NUMBER 0
58 #else
59 # ifdef AP_SERVER_MAJORVERSION_NUMBER
```

new/usr/src/lib/print/mod\_ipp/mod\_ipp.c

2

```
60 # define AP_SERVER_MAJORVERSION_NUMBER 2
61 # endif
62 # ifndef AP_SERVER_MINORVERSION_NUMBER
63 # define AP_SERVER_MINORVERSION_NUMBER 0
64 # endif
49 #ifndef APACHE_RELEASE /* appears to only exist in Apache 1.X */
50 #define APACHE2
51 #include "apr_compat.h"
65 #endif

67 #include <papi.h>
68 #include <ipp-listener.h>

70 #if AP_SERVER_MAJORVERSION_NUMBER < 2
57 #ifndef APACHE2
71 module MODULE_VAR_EXPORT ipp_module;
72 #else
73 module AP_MODULE_DECLARE_DATA ipp_module;
74 #endif

76 #ifndef AP_INIT_TAKE1 /* Apache 2.X has this, but 1.3.X does not */
77 #define AP_INIT_NO_ARGS(directive, action, arg, where, msg) \
78 { directive, action, arg, where, NO_ARGS, msg }
79 #define AP_INIT_TAKE1(directive, action, arg, where, msg) \
80 { directive, action, arg, where, TAKE1, msg }
81 #define AP_INIT_TAKE2(directive, action, arg, where, msg) \
82 { directive, action, arg, where, TAKE2, msg }
83 #endif

85 typedef struct {
86     int conformance;
87     char *default_user;
88     char *default_svc;
89     papi_attribute_t **operations;
90 } IPPListenerConfig;
    unchanged_portion_omitted
121 #endif

123 static ssize_t
124 read_data(void *fd, void *buf, size_t siz)
125 {
126     ssize_t len_read;
127     request_rec *ap_r = (request_rec *)fd;

129     len_read = ap_get_client_block(ap_r, buf, siz);
130 #if AP_SERVER_MAJORVERSION_NUMBER < 2
117 #ifndef APACHE2
131     ap_reset_timeout(ap_r);
132 #endif

134 #ifdef DEBUG
135     fprintf(stderr, "read_data(0x%8.8x, 0x%8.8x, %d): %d",
136             fd, buf, siz, len_read);
137     if (len_read < 0)
138         fprintf(stderr, ": %s", strerror(errno));
139     putc('\n', stderr);
140     dump_buffer(stderr, "read_data:", buf, len_read);
141 #endif

143     return (len_read);
144 }

146 static ssize_t
147 write_data(void *fd, void *buf, size_t siz)
148 {
149     ssize_t len_written;
```





```

277 #if AP_SERVER_MAJORVERSION_NUMBER >= 2
278 #ifdef APACHE2
279     ap_get_remote_host
280     (r->connection, r->per_dir_config, REMOTE_NAME, NULL));
281 #else
282     ap_get_remote_host
283     (r->connection, r->per_dir_config, REMOTE_NAME));
284 #endif
285
286     (void) papiAttributeListAddInteger(&request, PAPI_ATTR_EXCL,
287     "uri-port", ap_get_server_port(r));
288     if (r->headers_in != NULL) {
289 #if AP_SERVER_MAJORVERSION_NUMBER >= 2 || AP_SERVER_MINORVERSION_NUMBER > 2
290     char *host = (char *)apr_table_get(r->headers_in, "Host");
291 #else
292     char *host = (char *)ap_table_get(r->headers_in, "Host");
293 #endif
294 #endif /* ! codereview */
295
296     if ((host == NULL) || (host[0] == '\0'))
297     host = (char *)ap_get_server_name(r);
298
299     (void) papiAttributeListAddString(&request, PAPI_ATTR_EXCL,
300     "uri-host", host);
301 }
302 (void) papiAttributeListAddString(&request, PAPI_ATTR_EXCL,
303     "uri-path", r->uri);
304
305 config = ap_get_module_config(r->per_dir_config, &ipp_module);
306 if (config != NULL) {
307     (void) papiAttributeListAddInteger(&request, PAPI_ATTR_EXCL,
308     "conformance", config->conformance);
309     (void) papiAttributeListAddCollection(&request, PAPI_ATTR_EXCL,
310     "operations", config->operations);
311     if (config->default_user != NULL)
312     (void) papiAttributeListAddString(&request,
313     PAPI_ATTR_EXCL, "default-user",
314     config->default_user);
315     if (config->default_svc != NULL)
316     (void) papiAttributeListAddString(&request,
317     PAPI_ATTR_EXCL, "default-service",
318     config->default_svc);
319 }
320
321 /*
322  * For Trusted Solaris, pass the fd number of the socket connection
323  * to the backend so the it can be forwarded to the backend print
324  * service to retrieve the sensitivity label off of a multi-level
325  * service to retrieve the sensitativity label off of a multi-level
326  * port.
327  */
328 #if AP_SERVER_MAJORVERSION_NUMBER > 1
329 st = apr_os_sock_get(os_sock, r->connection->cs->pfd.desc.s);
330 if (st == APR_SUCCESS) {
331     (void) papiAttributeListAddInteger(&request, PAPI_ATTR_EXCL,
332     "peer-socket", (int)os_sock);
333 }
334 #else
335 #endif /* ! codereview */
336 (void) papiAttributeListAddInteger(&request, PAPI_ATTR_EXCL,
337     "peer-socket", ap_bfileno(r->connection->client, B_RD));
338 #endif /* ! codereview */
339
340 /* process the request */

```

```

341     status = ipp_process_request(request, &response, read_data, r);
342     if (status != PAPI_OK) {
343         errno = 0;
344         _log_rerror(APLOG_MARK, APLOG_ERR, r,
345         "request failed: %s\n", papiStatusString(status));
346         discard_data(r);
347     }
348 #ifdef DEBUG
349     fprintf(stderr, "processing result: %s\n", papiStatusString(status));
350     papiAttributeListPrint(stderr, response, "response (%d) ", getpid());
351 #endif
352
353 /*
354  * If the client is using chunking and we have not yet received the
355  * final "0" sized chunk, we need to discard any data that may
356  * remain in the post request.
357  */
358     if ((r->read_chunked != 0) &&
359     #if AP_SERVER_MAJORVERSION_NUMBER >= 2 || AP_SERVER_MINORVERSION_NUMBER > 2
360     (apr_table_get(r->headers_in, "Content-Length") == NULL))
361 #else
362     (ap_table_get(r->headers_in, "Content-Length") == NULL))
363 #endif
364 #endif /* ! codereview */
365 #endif /* ! codereview */
366     discard_data(r);
367
368 /* write an IPP response back to the network */
369     r->content_type = "application/ipp";
370
371 #if AP_SERVER_MAJORVERSION_NUMBER < 2
372 #ifndef APACHE2
373     ap_send_http_header(r);
374 #endif
375
376     status = ipp_write_message(write_data, r, response);
377     if (status != PAPI_OK)
378     _log_rerror(APLOG_MARK, APLOG_ERR, r,
379     "write failed: %s\n", papiStatusString(status));
380 #ifdef DEBUG
381     fprintf(stderr, "write result: %s\n", papiStatusString(status));
382     fflush(stderr);
383 #endif
384
385     papiAttributeListFree(request);
386     papiAttributeListFree(response);
387
388 #if AP_SERVER_MAJORVERSION_NUMBER < 2
389 #ifndef APACHE2
390     ap_kill_timeout(r);
391     if (ap_rflush(r) < 0)
392     _log_rerror(APLOG_MARK, APLOG_ERR, r,
393     "flush failed, response may not have been sent");
394 #endif
395 #endif
396
397     return (OK);
398 }
399
400 /*ARGSUSED1*/
401 static void *
402 create_ipp_dir_config(
403 #if AP_SERVER_MAJORVERSION_NUMBER < 2
404 #ifndef APACHE2
405     pool *p,
406 #else

```

```

404     apr_pool_t *p,
405 #endif
406     char *dirspec)
407 {
408     IPPListenerConfig *config =
409 #if AP_SERVER_MAJORVERSION_NUMBER < 2
280 #ifndef APACHE2
410         ap_palloc(p, sizeof (*config));
411 #else
412         apr_palloc(p, sizeof (*config));
413 #endif
414
415     if (config != NULL) {
416         (void) memset(config, 0, sizeof (*config));
417         config->conformance = IPP_PARSE_CONFORMANCE_RASH;
418         config->default_user = NULL;
419         config->default_svc = NULL;
420         (void) ipp_configure_operation(&config->operations, "required",
421                                     "enable");
422     }
423
424     return (config);
425 }

```

unchanged portion omitted

```

446 /*ARGSUSED0*/
447 static const char *
448 ipp_operation(cmd_parms *cmd, void *cfg, const char *op, const char *toggle)
319 ipp_operation(cmd_parms *cmd, void *cfg, char *op, char *toggle)
449 {
450     IPPListenerConfig *config = (IPPListenerConfig *)cfg;
451     papi_status_t status;
452
453     status = ipp_configure_operation(&config->operations, op, toggle);
454     switch (status) {
455     case PAPI_OK:
456         return (NULL);
457     case PAPI_BAD_ARGUMENT:
458         return (gettext("internal error (invalid argument)"));
459     default:
460         return (papiStatusString(status));
461     }
462
463     /* NOTREACHED */
464     /* return (gettext("contact your software vendor")); */
465 }

```

unchanged portion omitted

```

518 #if AP_SERVER_MAJORVERSION_NUMBER >= 2
389 #ifndef APACHE2
519 /*ARGSUSED0*/
520 static const char *
521 ipp_method(const request_rec *r)
522 {
523     return ("ipp");
524 }

```

unchanged portion omitted

```

533 /* Dispatch list for API hooks */
534 /*ARGSUSED0*/
535 static void
536 ipp_register_hooks(apr_pool_t *p)
537 {
538     static const char * const modules[] = { "mod_dir.c", NULL };
539
540     /* Need to make sure we don't get directory listings by accident */

```

```

541     ap_hook_handler(ipp_handler, NULL, modules, APR_HOOK_MIDDLE);
542     ap_hook_default_port(ipp_port, NULL, NULL, APR_HOOK_MIDDLE);
543 #if AP_SERVER_MAJORVERSION_NUMBER >= 2
544     ap_hook_http_scheme(ipp_method, NULL, NULL, APR_HOOK_MIDDLE);
545 #else
546 #endif /* ! codereview */
547     ap_hook_http_method(ipp_method, NULL, NULL, APR_HOOK_MIDDLE);
548 #endif
549 #endif /* ! codereview */
550 }

```

```

552 module AP_MODULE_DECLARE_DATA ipp_module = {
553     STANDARD20_MODULE_STUFF,
554     create_ipp_dir_config,          /* create per-dir    config    */
555     NULL,                          /* merge per-dir    config    */
556     NULL,                          /* create per-server config */
557     NULL,                          /* merge per-server config */
558     ipp_cmds,                      /* table of config  commands */
559     ipp_register_hooks             /* register hooks    */
560 };

```

```

562 #else /* Apache 1.X */

```

```

564 /* Dispatch list of content handlers */
565 static const handler_rec ipp_handlers[] = {
566     /*
567     * This handler association causes all IPP request with the
568     * correct MIME type to call the protocol handler.
569     */
570     { "application/ipp", ipp_handler },
571     /*
572     * This handler association is causes everything to go through the IPP
573     * protocol request handler. This is necessary because client POST
574     * request may be for something outside of the normal printer-uri
575     * space.
576     */
577     { "**/*", ipp_handler },
578
579     { NULL, NULL }
580 };

```

```

583 module MODULE_VAR_EXPORT ipp_module = {
584     STANDARD_MODULE_STUFF,
585     NULL,                          /* module initializer */
586     create_ipp_dir_config,          /* create per-dir    config structures */
587     NULL,                          /* merge per-dir    config structures */
588     NULL,                          /* create per-server config structures */
589     NULL,                          /* merge per-server config structures */
590     ipp_cmds,                      /* table of config  file commands */
591     ipp_handlers,                  /* [#8] MIME-typed-dispatched handlers */
592     NULL,                          /* [#1] URI to filename translation */
593     NULL,                          /* [#4] validate user id from request */
594     NULL,                          /* [#5] check if the user is ok_here */
595     NULL,                          /* [#3] check access by host address */
596     NULL,                          /* [#6] determine MIME type */
597     NULL,                          /* [#7] pre-run fixups */
598     NULL,                          /* [#9] log a transaction */
599     NULL,                          /* [#2] header parser */
600     NULL,                          /* child_init */
601     NULL,                          /* child_exit */
602     NULL,                          /* [#0] post read-request */
603 };
604 #endif

```

\*\*\*\*\*

23795 Wed May 7 18:40:55 2014

new/usr/src/man/man1m/logadm.1m

2837 - remove print/lp\* from gate and use CUPS from userland

\*\*\*\*\*

```

1 \" te
2.\" Copyright (c) 2007 Sun Microsystems, Inc. All Rights Reserved.
3.\" The contents of this file are subject to the terms of the Common Development
4.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
5.\" When distributing Covered Code, include this CDDL HEADER in each file and in
6.TH LOGADM 1M "May 23, 2007"
7.SH NAME
8 logadm \- manage endlessly growing log files
9.SH SYNOPSIS
10.LP
11.nf
12 \fBlogadm\fR
13.fi

15.LP
16.nf
17 \fBlogadm\fR [\fB-\fR\fR \fR] \fR \fR
18.fi

20.SH DESCRIPTION
21.sp
22.LP
23 \fBlogadm\fR is a general log rotation tool that is suitable for running from
24 \fBcron\fR(1M).
25.sp
26.LP
27 Without arguments, \fBlogadm\fR reads the \fB/etc/logadm.conf\fR file, and for
28 every entry found in that file checks the corresponding log file to see if it
29 should be rotated. Typically this check is done each morning by an entry in the
30 root's \fBcrontab\fR.
31.sp
32.LP
33 If the \fR argument is specified, \fBlogadm\fR renames the
34 corresponding log file by adding a suffix so that the most recent log file ends
35 with \fR (that is, \fR, the next most recent ends
36 with \fR (that is, \fR, and so forth. By default,
37 ten versions of old log files are kept (that is, \fR
38 through \fR). At the point when what would be the eleventh
39 file is logged, \fBlogadm\fR automatically deletes the oldest version to keep
40 the count of files at ten.
41.sp
42.LP
43 \fBlogadm\fR takes a number of \fR. You can specify these options on
44 the command line or in the \fB/etc/logadm.conf\fR file. The \fBlogadm\fR
45 command searches \fB/etc/logadm.conf\fR for lines of the form \fR
46 \fR
47.sp
48.ne 2
49.na
50 \fB\fR
51.ad
52.sp .6
53.RS 4n
54 Identifies an entry in \fB/etc/logadm.conf\fR. This can be a name or the
55 pathname of the log file. If you specify a log file, rather than a name, for
56 this field, it must be a fully qualified pathname.
57.RE

59.sp
60.ne 2
61.na

```

```

62 \fR
63.ad
64.sp .6
65.RS 4n
66 Identifies command line options exactly as they would be entered on the command
67 line. This allows commonly used log rotation policies to be stored in the
68 \fB/etc/logadm.conf\fR file. See \fBEXAMPLES\fR.
69.sp
70 If \fR are specified both in \fB/etc/logadm.conf\fR and on the
71 command line, those in the \fB/etc/logadm.conf\fR file are applied first.
72 Therefore, the command line options override those in \fB/etc/logadm.conf\fR.
73.sp
74 Log file names specified in \fB/etc/logadm.conf\fR may contain filename
75 substitution characters such as \fR and \fR, that are supported by
76 \fBcsh\fR(1).
77.RE

79.sp
80.LP
81 Two options control when a log file is rotated. They are: \fR size
82 \fR period.
83.sp
84.LP
85 When using more than one of these options at a time, there is an implied
86 \fR between them. This means that all conditions must be met before the
87 log is rotated.
88.sp
89.LP
90 If neither of these two options are specified, the default conditions for
91 rotating a log file are: \fR \fR \fR, which means
92 the log file is only rotated if the size is non-zero and if at least 1 week has
93 passed since the last time it was rotated.
94.sp
95.LP
96 By specifying \fR as a rotation condition, any other
97 rotation conditions are ignored and \fBlogadm\fR moves on to the expiration of
98 old log files. By specifying \fR as a rotation condition,
99 a log rotation is forced.
100.sp
101.LP
102 Unless specified by the \fR, \fR, or \fR options, \fBlogadm\fR
103 replaces the log file (after renaming it) by creating an empty file whose
104 owner, group ID, and permissions match the original file.
105.sp
106.LP
107 Three options control when old log files are expired: \fR age \fR
108 count \fR size. These options expire the oldest log files until a
109 particular condition or conditions are met. For example, the combination
110 \fR and the \fR options expires old log
111 files until there are no more than 5 of the \fR their combined disk usage
112 is no more than 10 megabytes. If none of these options are specified, the
113 default expiration is \fR \fR which keeps ten old log files. If
114 no files are to be expired, use \fR to prevent expiration by
115 default.
116.SH OPTIONS
117.sp
118.LP
119 The following options are supported:
120.sp
121.ne 2
122.na
123 \fB\fR \fR
124.ad
125.sp .6
126.RS 4n
127 Execute the \fR after renaming the log file. \fR

```

```

128 is passed to \fBsh\fR \fB-c\fR.
129 .sp
130 Specify \fIpost_command\fR as a valid shell command. Use quotes to protect
131 spaces or shell metacharacters in \fIpost_command\fR.
132 .sp
133 This option can be used to restart a daemon that is writing to the file. When
134 rotating multiple logs with one \fBlogadm\fR command, \fIpost_command\fR is
135 executed only once after all the logs are rotated, not once per rotated log.
136 .RE

138 .sp
139 .ne 2
140 .na
141 \fB\fB-A\fR \fIage\fR\fR
142 .ad
143 .sp .6
144 .RS 4n
145 Delete any versions that have not been modified for the amount of time
146 specified by \fIage\fR.
147 .sp
148 Specify \fIage\fR as a number followed by an \fBh\fR (hours), \fBd\fR (days),
149 \fBw\fR (weeks), \fBm\fR (months), or \fBy\fR (years).
150 .RE

152 .sp
153 .ne 2
154 .na
155 \fB\fB-b\fR \fIpre_command\fR\fR
156 .ad
157 .sp .6
158 .RS 4n
159 Execute \fIpre_command\fR before renaming the log file. \fIpre_command\fR is
160 passed to \fBsh\fR \fB-c\fR.
161 .sp
162 Specify \fIpre_command\fR as a valid shell command. Use quotes to protect
163 spaces or shell metacharacters in the \fIpre_command\fR.
164 .sp
165 This option can be used to stop a daemon that is writing to the file. When
166 rotating multiple logs with one \fBlogadm\fR command, \fIpre_command\fR is
167 executed only once before all the logs are rotated, not once per rotated log.
168 .RE

170 .sp
171 .ne 2
172 .na
173 \fB\fB-c\fR\fR
174 .ad
175 .sp .6
176 .RS 4n
177 Rotate the log file by copying it and truncating the original logfile to zero
178 length, rather than renaming the file.
179 .RE

181 .sp
182 .ne 2
183 .na
184 \fB\fB-C\fR \fIcount\fR\fR
185 .ad
186 .sp .6
187 .RS 4n
188 Delete the oldest versions until there are not more than \fIcount\fR files
189 left.
190 .sp
191 If no expire options (\fB-A\fR, \fB-C\fR, or \fB-S\fR) are specified,
192 \fB\fR\fB-C\fR \fB10\fR is the default. To prevent the default expire rule from
193 being added automatically, specify \fB\fR\fB-C\fR \fB0\fR .

```

```

194 .RE

196 .sp
197 .ne 2
198 .na
199 \fB\fB-e\fR \fIemail_addr\fR\fR
200 .ad
201 .sp .6
202 .RS 4n
203 Send error messages by email to \fIemail_addr\fR.
204 .sp
205 As \fBlogadm\fR is typically run from \fBcron\fR(1M), error messages are
206 captured by \fBcron\fR and mailed to the owner of the \fBcrontab\fR.
207 .sp
208 This option is useful if you want the mail regarding error messages to go to
209 another address instead. If no errors are encountered, no mail message is
210 generated.
211 .RE

213 .sp
214 .ne 2
215 .na
216 \fB\fB-E\fR \fIcmd\fR\fR
217 .ad
218 .sp .6
219 .RS 4n
220 Execute \fIcmd\fR to expire the file, rather than deleting the old log file to
221 expire it.
222 .sp
223 \fIcmd\fR is passed it to \fBsh\fR \fB-c\fR. The file is considered expired
224 after \fIcmd\fR completes. If the old log file is not removed or renamed by the
225 \fIcmd\fR, \fBlogadm\fR considers it for expiration the next time that it runs
226 on the specified log file. If present, the keyword \fB$file\fR is expanded in
227 the specified \fIcmd\fR to the name of the file being expired.
228 .sp
229 This option is useful for tasks such as mailing old log files to
230 administrators, or copying old log files to long term storage.
231 .RE

233 .sp
234 .ne 2
235 .na
236 \fB\fB-f\fR \fIconf_file\fR\fR
237 .ad
238 .sp .6
239 .RS 4n
240 Use \fIconf_file\fR instead of \fB/etc/logadm.conf\fR.
241 .sp
242 This option allows non-root users to keep their own \fBlogadm\fR configuration
243 files.
244 .RE

246 .sp
247 .ne 2
248 .na
249 \fB\fB-g\fR \fIgroup\fR\fR
250 .ad
251 .sp .6
252 .RS 4n
253 Create a new empty file with the \fBID\fR specified by \fIgroup\fR, instead of
254 preserving the group \fBID\fR of the log file.
255 .sp
256 Specify \fIgroup\fR by name or by numeric group \fBID\fR, as accepted by
257 \fBchgrp\fR(1).
258 .sp
259 This option requires the ability to change file group ownership using the

```

```

260 \fBchgrp\fR(1) command.
261 .RE

263 .sp
264 .ne 2
265 .na
266 \fB\fB-h\fR\fR
267 .ad
268 .sp .6
269 .RS 4n
270 Print a help message that describes \fBblogadm\fR's options.
271 .RE

273 .sp
274 .ne 2
275 .na
276 \fB\fB-l\fR\fR
277 .ad
278 .sp .6
279 .RS 4n
280 Use local time rather than the Coordinated Universal Time (UTC) when naming
281 rotated log files (see the discussion of percent sequences in the templates
282 supplied with the \fB-t\fR option).
283 .RE

285 .sp
286 .ne 2
287 .na
288 \fB\fB-m\fR \fImode\fR
289 .ad
290 .sp .6
291 .RS 4n
292 Create a new empty file with the mode specified by \fImode\fR, instead of
293 preserving the mode of the log file.
294 .sp
295 Specify \fImode\fR in any form that is accepted by the \fBchmod\fR(1) command.
296 .RE

298 .sp
299 .ne 2
300 .na
301 \fB\fB-M\fR \fIcmd\fR
302 .ad
303 .sp .6
304 .RS 4n
305 Use \fIcmd\fR to rename the log file. If the keyword \fB$file\fR is specified,
306 it is expanded to the name of the log file. Similarly, the keyword \fB$file\fR
307 is expanded to the new name of the log file. The \fB$file\fR keyword is only
308 available with commands provided with the \fB-M\fR option. After the command
309 completes, the log file is replaced by the rotate file. The default \fIcmd\fR
310 is "\fB/bin/mv\fR \fB$file\fR \fB$file\fR".
311 .RE

313 .sp
314 .ne 2
315 .na
316 \fB\fB-n\fR \fI\fR
317 .ad
318 .sp .6
319 .RS 4n
320 Print the actions that the \fBblogadm\fR command will perform without actually
321 performing them.
322 .sp
323 This option is useful for checking arguments before making any changes to the
324 system.
325 .sp

```

```

326 It is important to remember, however, that since log rotating actions are only
327 printed with this option, \fBblogadm\fR might not find files that need expiring,
328 but if run without the \fB-n\fR \fBblogadm\fR might create a file that needs
329 expiring by performing the log rotating actions. Therefore, if you see no files
330 being expired with the \fB-n\fR option, files still might be expired without
331 it.
332 .RE

334 .sp
335 .ne 2
336 .na
337 \fB\fB-N\fR\fR
338 .ad
339 .sp .6
340 .RS 4n
341 Prevent an error message if the specified logfile does not exist. Normally,
342 \fBblogadm\fR produces an error message if the log file is not found. With
343 \fB-N\fR, if the log file doesn't exist \fBblogadm\fR moves on to the expire
344 rules (if any) and then to the next log file (if any), without creating the
345 empty replacement log file.
346 .RE

348 .sp
349 .ne 2
350 .na
351 \fB\fB-o\fR \fIowner\fR
352 .ad
353 .sp .6
354 .RS 4n
355 Create the new empty file with \fIowner\fR, instead of preserving the owner of
356 the log file.
357 .sp
358 Specify \fIowner\fR in any form that is accepted by the \fBchown\fR(1) command.
359 .RE

361 .sp
362 .ne 2
363 .na
364 \fB\fB-p\fR \fIperiod\fR
365 .ad
366 .sp .6
367 .RS 4n
368 Rotate a log file after the specified time period (\fIperiod\fR).
369 .sp
370 Specify \fIperiod\fR as a number followed by \fBd\fR for days, \fBh\fR for
371 hours, \fBw\fR for weeks, \fBm\fR for months (30 days) or \fBy\fR for years.
372 There are also two special values for period: \fBnow\fR and \fBnever\fR.
373 "\fB\fB-p\fR \fBnow\fR" forces log rotation. "\fB\fB-p\fR \fBnever\fR"
374 forces no log rotation.
375 .RE

377 .sp
378 .ne 2
379 .na
380 \fB\fB-P\fR \fItimestamp\fR
381 .ad
382 .sp .6
383 .RS 4n
384 Used by \fBblogadm\fR to record the last time the log was rotated in
385 \fB/etc/logadm.conf\fR.
386 .sp
387 This option uses \fItimestamp\fR to determine if the log rotation period has
388 passed. The format of \fItimestamp\fR matches the format generated by
389 \fBctime\fR(3C), with quotes around it to protect embedded spaces.
390 \fItimestamp\fR is always recorded in the Coordinated Universal Time (UTC)
391 timezone.

```

```

392 .RE
394 .sp
395 .ne 2
396 .na
397 \fB\fB-r\fR\fR
398 .ad
399 .sp .6
400 .RS 4n
401 Remove any entries corresponding to the specified \fIlogname\fR from the
402 \fB/etc/logadm.conf\fR.
403 .RE
405 .sp
406 .ne 2
407 .na
408 \fB\fB-R\fR \fIcmd\fR
409 .ad
410 .sp .6
411 .RS 4n
412 Run the \fIcmd\fR when an old log file is created by a log rotation. If the
413 keyword \fB$file\fR is embedded in the specified command, it is expanded to the
414 name of the old log file just created by log rotation.
415 .sp
416 This option is useful for processing log file contents after rotating the log.
417 \fIcmd\fR is executed by passing it to \fBsh\fR \fB-c\fR. When rotating
418 multiple logs with one logadm command, the command supplied with \fB-R\fR is
419 executed once every time a log is rotated. This is useful for post-processing a
420 log file (that is, sorting it, removing uninteresting lines, etc.). The
421 \fB-a\fR option is a better choice for restarting daemons after log rotation.
422 .RE
424 .sp
425 .ne 2
426 .na
427 \fB\fB-s\fR \fIsize\fR
428 .ad
429 .sp .6
430 .RS 4n
431 Rotate the log file only if its size is greater than or equal to \fIsize\fR.
432 .sp
433 Specify \fIsize\fR as a number followed by the letter \fBb\fR for bytes,
434 \fBk\fR for kilobytes, \fBm\fR for megabytes, or \fBg\fR for gigabytes.
435 .RE
437 .sp
438 .ne 2
439 .na
440 \fB\fB-S\fR \fIsize\fR
441 .ad
442 .sp .6
443 .RS 4n
444 Delete the oldest versions until the total disk space used by the old log files
445 is less than the specified size.
446 .sp
447 Specify \fIsize\fR as a number followed by the letter \fBb\fR for bytes,
448 \fBk\fR for kilobytes, \fBm\fR for megabytes, or \fBg\fR for gigabytes.
449 .RE
451 .sp
452 .ne 2
453 .na
454 \fB\fB-t\fR \fItemplate\fR
455 .ad
456 .sp .6
457 .RS 4n

```

```

458 Specify the template to use when renaming log files.
459 .sp
460 \fItemplate\fR can be a simple name, such as \fB/var/adm/oldfile\fR, or it can
461 contain special keywords which are expanded by \fBlogadm\fR and are in the form
462 \fB$\fR\fIword\fR. Allowed sequences are:
463 .sp
464 .ne 2
465 .na
466 \fB$\fIbasename\fR
467 .ad
468 .sp .6
469 .RS 4n
470 The log file name, without the directory name
471 .RE
473 .sp
474 .ne 2
475 .na
476 \fB$\fIdirname\fR
477 .ad
478 .sp .6
479 .RS 4n
480 The directory of the file to be rotated
481 .RE
483 .sp
484 .ne 2
485 .na
486 \fB$\fIdomain\fR
487 .ad
488 .sp .6
489 .RS 4n
490 Expands to the output of \fBdomainname\fR
491 .RE
493 .sp
494 .ne 2
495 .na
496 \fB$\fIfile\fR
497 .ad
498 .sp .6
499 .RS 4n
500 The full path name of the file to be rotated
501 .RE
503 .sp
504 .ne 2
505 .na
506 \fB$\fIisa\fR
507 .ad
508 .sp .6
509 .RS 4n
510 Expands to the output of \fBuname\fR \fB-p\fR
511 .RE
513 .sp
514 .ne 2
515 .na
516 \fB$\fImachine\fR
517 .ad
518 .sp .6
519 .RS 4n
520 Expands to the output of \fBuname\fR \fB-m\fR
521 .RE
523 .sp

```

```

524 .ne 2
525 .na
526 \fB$\fIn\fR\fR
527 .ad
528 .sp .6
529 .RS 4n
530 The version number, \fB0\fR is most recent, \fB1\fR is next most recent, and so
531 forth
532 .RE

534 .sp
535 .ne 2
536 .na
537 \fB$\fIN\fR\fR
538 .ad
539 .sp .6
540 .RS 4n
541 The same as $\fIn\fR, but starts at \fB1\fR instead of zero
542 .RE

544 .sp
545 .ne 2
546 .na
547 \fB$\fInodename\fR\fR
548 .ad
549 .sp .6
550 .RS 4n
551 Expands to the output of \fBuname\fR \fB-n\fR
552 .RE

554 .sp
555 .ne 2
556 .na
557 \fB$\fIplatform\fR\fR
558 .ad
559 .sp .6
560 .RS 4n
561 Expands to the output of \fBuname\fR \fB-i\fR
562 .RE

564 .sp
565 .ne 2
566 .na
567 \fB$\fIrelease\fR\fR
568 .ad
569 .sp .6
570 .RS 4n
571 Expands to the output of \fBuname\fR \fB-r\fR
572 .RE

574 .sp
575 .ne 2
576 .na
577 \fB$\fIsecs\fR\fR
578 .ad
579 .sp .6
580 .RS 4n
581 The number of seconds since \fB00:00:00 UTC, January 1,1970\fR
582 .RE

584 .sp
585 .ne 2
586 .na
587 \fB$\fIzonename\fR\fR
588 .ad
589 .sp .6

```

```

590 .RS 4n
591 Expands to the output of \fBzonename\fR(1).
592 .RE

594 To actually have the dollar sign character in the file name, use \fB$$\fR. Any
595 percent sequences allowed by \fBstrftime\fR(3C) are also allowed, for example,
596 \fB%d\fR expands to the day of the month. To actually have a percent sign
597 character in the file name, use \fB%\fR. Both dollar-sign keywords and percent
598 sequences can appear anywhere in the template. If the template results in a
599 pathname with non-existent directories, they are created as necessary when
600 rotating the log file.
601 .sp
602 If no \fB-t\fR option is specified, the default template is \fB$file.$n\fR.
603 Actual \fBrotation\fR of log files, where each version is shifted up until it
604 expires is done using the \fB$n\fR keyword. If the template does not contain
605 the \fB$n\fR keyword, the log file is simply renamed to the new name and then
606 the expire rules, if any, are applied.
607 .RE

609 .sp
610 .ne 2
611 .na
612 \fB\fB-T\fR \fIpattern\fR\fR
613 .ad
614 .sp .6
615 .RS 4n
616 Normally \fBblogadm\fR looks for a list of old log files by turning the template
617 (specified with the \fB-t\fR option) into a pattern and finding existing files
618 whose names match that pattern. The \fB-T\fR option causes the given pattern to
619 be used instead.
620 .sp
621 This option is useful if another program fiddles with the old log file names,
622 like a \fBcron\fR job to compress them over time. The pattern is in the form of
623 a pathname with special characters such as \fB*\fR and \fB?\fR as supported by
624 \fBcsh\fR(1) filename substitution.
625 .RE

627 .sp
628 .ne 2
629 .na
630 \fB\fB-v\fR\fR
631 .ad
632 .sp .6
633 .RS 4n
634 Print information about the actions being executed in verbose mode.
635 .RE

637 .sp
638 .ne 2
639 .na
640 \fB\fB-V\fR\fR
641 .ad
642 .sp .6
643 .RS 4n
644 Validate the configuration file.
645 .sp
646 This option validates that an entry for the specified \fBillogname\fR exists in
647 the \fB/etc/logadm.conf\fR file and is syntactically correct. If \fBillogname\fR
648 is not specified, all entries in the configuration file are validated. If a
649 \fBillogname\fR argument is specified, the command validates the syntax of that
650 entry. If the entry is found, it is printed and the exit value of the command
651 is true. Otherwise the exit value is false.
652 .RE

654 .sp
655 .ne 2

```

```

656 .na
657 \fB\fB-w\fR \fIentryname\fR\fR
658 .ad
659 .sp .6
660 .RS 4n
661 Write an entry into the config file (that is, \fB/etc/logadm.conf\fR) that
662 corresponds to the current command line arguments. If an entry already existed
663 for the specified \fIentryname\fR, it is removed first. This is the preferred
664 method for updating \fB/etc/logadm.conf\fR, because it prevents syntax errors.
665 The \fIentryname\fR is an argument to an invocation of \fBblogadm\fR.
666 \fIentryname\fR might be chosen as something easy to remember or it can be the
667 pathname of the log file. If a pathname, rather than a name is used, it must be
668 a fully qualified pathname.
669 .sp
670 If no log file name is provided on a \fBblogadm\fR command line, the entry name
671 is assumed to be the same as the log file name. For example, the following two
672 lines achieve the same thing, keeping two copies of rotated log files:
673 .sp
674 .in +2
675 .nf
676 % logadm -C2 -w mylog /my/really/long/log/file/name
677 % logadm -C2 -w /my/really/long/log/file/name
678 .fi
679 .in -2
680 .sp

682 .RE

684 .sp
685 .ne 2
686 .na
687 \fB\fB-z\fR \fIcount\fR\fR
688 .ad
689 .sp .6
690 .RS 4n
691 Compress old log files after all other commands have been executed. \fIcount\fR
692 of the most recent log files are left uncompressed, therefore making the
693 \fIcount\fR most recent files easier to peruse. Use \fIcount\fR of zero to
694 compress all old logs.
695 .sp
696 The compression is done with \fBgzip\fR(1) and the resulting log file has the
697 suffix of \fB\&.gz\fR.
698 .RE

700 .SH OPERANDS
701 .sp
702 .LP
703 The following operands are supported:
704 .sp
705 .ne 2
706 .na
707 \fB\fIlogname\fR\fR
708 .ad
709 .sp .6
710 .RS 4n
711 Identifies the name of the entry in \fB/etc/logadm.conf\fR. If the log file
712 name is specified in the \fIlogname\fR field, it is assumed that \fIlogname\fR
713 is the same as the actual log file name.
714 .RE

716 .SH EXAMPLES
717 .LP
718 \fBExample 1\fR \fRRotating a File and Keeping Previous Versions
719 .sp
720 .LP
721 The following example rotates the \fB/var/adm/exacct/proc\fR file, keeping ten

```

```

722 previous versions in \fB/var/adm/exacct/proc.0\fR through
723 \fB/var/adm/exacct/proc.9\fR.

725 .sp
726 .LP
727 Tell \fBblogadm\fR to copy the file and truncate it.

729 .sp
730 .in +2
731 .nf
732 % logadm -c /var/adm/exacct/proc
733 .fi
734 .in -2
735 .sp

737 .LP
738 \fBExample 2\fR \fRRotating syslog
739 .sp
740 .LP
741 The following example rotates \fBsyslog\fR and keeps eight log files. Old log
742 files are put in the directory \fB/var/oldlogs\fR instead of \fB/var/log\fR:

744 .sp
745 .in +2
746 .nf
747 % logadm -C8 -t'/var/oldlogs/syslog.$n' /var/log/syslog
748 .fi
749 .in -2
750 .sp

752 .LP
753 \fBExample 3\fR \fRRotating /var/adm/sulog and Expiring Based on Age
754 .sp
755 .LP
756 The following entry in the \fB/etc/logadm.conf\fR file rotates the
757 \fB/var/adm/sulog\fR file and expires any copies older than 30 days.

759 .sp
760 .in +2
761 .nf
762 /var/adm/sulog -A 30d
763 .fi
764 .in -2
765 .sp

767 .LP
768 \fBExample 4\fR \fRRotating Files and Expiring Based on Disk Usage
769 .sp
770 .LP
771 The following entry in the \fB/etc/logadm.conf\fR file rotates the
772 \fB/var/adm/sulog\fR file and expires old log files when more than 100
773 megabytes are used by the sum of all the rotated log files.

775 .sp
776 .in +2
777 .nf
778 /var/adm/sulog -S 100m
779 .fi
780 .in -2
781 .sp

783 .LP
784 \fBExample 5\fR \fRCreating an Entry that Stores the Logfile Name
785 .sp
786 .LP
787 This example creates an entry storing the log file name and the fact that we

```



```

788 want to keep 20 copies in \fB/etc/logadm.conf\fR, but the \fB\fR\fB-p\fR
789 \fBnever\fR means the entry is ignored by the normal \fBblogadm\fR run from
790 root's crontab every morning.

792 .sp
793 .in +2
794 .nf
795 % logadm -w locallog /usr/local/logfile -C20 -p never
796 .fi
797 .in -2
798 .sp

800 .sp
801 .LP
802 Use the following entry on the command line to override the \fB\fR\fB-p\fR
803 \fBnever\fR option:

805 .sp
806 .in +2
807 .nf
808 % logadm -p now locallog
809 .fi
810 .in -2
811 .sp

813 .LP
814 \fBExample 6 \fRRotating the apache Error and Access Logs
815 .sp
816 .LP
817 The following example rotates the apache error and access logs monthly to
818 filenames based on current year and month. It keeps the 24 most recent copies
819 and tells apache to restart after renaming the logs.

821 .sp
822 .LP
823 This command is run once, and since the \fB-w\fR option is specified, an entry
824 is made in \fB/etc/logadm.conf\fR so the apache logs are rotated from now on.

826 .sp
827 .in +2
828 .nf
829 % logadm -w apache -p 1m -C 24\`e
830     -t '/var/apache2/2.2/old-logs/$basename.%Y-%m'\`e
831     -a '/usr/apache2/2.2/bin/apachectl graceful'\`e
832     '/var/apache2/2.2/logs/*{access,error}_log'
830     -t '/var/apache/old-logs/$basename.%Y-%m'\`e
831     -a '/usr/apache/bin/apachectl graceful'\`e
832     '/var/apache/logs/*{access,error}_log'
833 .fi
834 .in -2
835 .sp

837 .sp
838 .LP
839 This example also illustrates that the entry name supplied with the \fB-w\fR
840 option doesn't have to match the log file name. In this example, the entry name
841 is \fBapache\fR and once the line has been run, the entry in
842 \fB/etc/logadm.conf\fR can be forced to run by executing the following command:

844 .sp
845 .in +2
846 .nf
847 % logadm -p now apache
848 .fi
849 .in -2
850 .sp

```

```

852 .sp
853 .LP
854 Because the expression matching the apache log file names was enclosed in
855 quotes, the expression is stored in \fB/etc/logadm.conf\fR, rather than the
856 list of files that it expands to. This means that each time \fBblogadm\fR runs
857 from \fBcron\fR it expands that expression and checks all the log files in the
858 resulting list to see if they need rotating.

860 .sp
861 .LP
862 The following command is an example without the quotes around the log name
863 expression. The shell expands the last argument into a list of log files that
864 exist at the time the command is entered, and writes an entry to
865 \fB/etc/logadm.conf\fR that rotates the files.

867 .sp
868 .in +2
869 .nf
870 logadm -w apache /var/apache2/2.2/logs/*_log
870 logadm -w apache /var/apache/logs/*_log
871 .fi
872 .in -2
873 .sp

875 .SH FILES
876 .sp
877 .ne 2
878 .na
879 \fB\fB/etc/logadm.conf\fR\fR
880 .ad
881 .sp .6
882 .RS 4n
883 configuration file for \fBblogadm\fR command
884 .RE

886 .SH ATTRIBUTES
887 .sp
888 .LP
889 See \fBattributes\fR(5) for descriptions of the following attributes:
890 .sp

892 .sp
893 .TS
894 box;
895 c | c
896 l | l .
897 ATTRIBUTE TYPE    ATTRIBUTE VALUE
898 -
899 Interface Stability    Evolving
900 .TE

902 .SH SEE ALSO
903 .sp
904 .LP
905 \fBbchgrp\fR(1), \fBbchmod\fR(1), \fBbchown\fR(1), \fBbcsh\fR(1), \fBbgzip\fR(1),
906 \fBbcron\fR(1M), \fBbctime\fR(3C), \fBbstrftime\fR(3C), \fBblogadm.conf\fR(4),
907 \fBattributes\fR(5)
908 .SH NOTES
909 .sp
910 .LP
911 When \fBblogadm\fR applies expire conditions (supplied by the \fB-A\fR,
912 \fB-C\fR, and \fB-S\fR options), it deletes files, the oldest first, until the
913 conditions are satisfied. If the template used for naming the old logs
914 contained \fB$n\fR or \fB$N\fR, \fBblogadm\fR picks the highest value of
915 \fB$n\fR or \fB$N\fR found in the old log file names first. If the template

```

916 used is something else, \fBlogadm\fR uses the modification time to determine  
917 which files to expire first. This may not be the expected behavior if an old  
918 log file has been modified since it was rotated.  
919 .sp  
920 .LP  
921 Depending on log file sizes and number of log files, log file rotations can be  
922 very time-consuming.  
923 .sp  
924 .LP  
925 By default, \fBlogadm\fR works in \fBGMT\fR. Therefore, all entries written to  
926 the \fB/etc/logadm.conf\fR file (see \fBlogadm.conf\fR(4)) will have a  
927 \fBGMT\fR timestamp. Users can use the \fB-l\fR option to set \fBlogadm\fR to  
928 local time.

```

*****
9243 Wed May 7 18:40:58 2014
new/usr/src/man/man4/mod_ipp.4
2837 - remove print/lp* from gate and use CUPS from userland
*****
1 \" te
2 .\" Copyright (c) 2003, Sun Microsystems, Inc.
3 .\" All Rights Reserved.
4 .\" The contents of this file are subject to the terms of the Common Development
5 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
6 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
7 .TH MOD_IPP 4 "Nov 26, 2003"
8 .SH NAME
9 mod_ipp \- Embedded Internet Print Protocol (IPP) listener for the Apache HTTP
10 server
11 .SH SYNOPSIS
12 .LP
13 .nf
14 \fBusr/apache2/2.2/libexec/mod_ipp.so\fR
15 \fBusr/apache/libexec/mod_ipp.so\fR
16 .fi

17 .SH DESCRIPTION
18 .sp
19 .LP
20 The \fBmod_ipp\fR module implements RFCs 2910 and 2911 to provide an IPP
21 handling service for the Apache HTTP server. When loaded on the Apache server,
22 \fBmod_ipp\fR processes all HTTP requests with MIME types of
23 \fBapplication/ipp\fR. The \fBmod_ipp\fR module also processes additional
24 configuration directives to enable or disable portions of the protocol support.
25 .SS "Using Configuration Directives"
26 .sp
27 .LP
28 The following is a list of configuration directives that apply to the Apache
29 IPP listening service:
30 .RS +4
31 .TP
32 .ie t \(\bu
33 .el o
34 ipp-conformance (\fIautomatic\fR|\fI1.0\fR|\fI1.1\fR)
35 .RE
36 .RS +4
37 .TP
38 .ie t \(\bu
39 .el o
40 ipp-operation (\fIoperation\fR) (\fBenable\fR|\fBdisable\fR)
41 .RS +4
42 .TP
43 .ie t \(\bu
44 .el o
45 \fBenable\fR|\fBdisable\fR
46 .sp
47 The values \fBtrue\fR, \fBbytes\fR, \fBbon\fR, \fBenable\fR are considered to be
48 synonymous and will enable support for the named operation. All other values
49 will disable support for the named operation.
50 .RE
51 .RE
52 .SS "Operations"
53 .sp
54 .LP
55 The following is a list of IPP handling service operations:
56 .sp
57 .ne 2
58 .na
59 \fBprint-job\fR
60 .ad

```

```

61 .RS 26n
62 This operation is a required IPP operation that allows client systems to submit
63 a print job with a single document embedded in the data stream. This operation
64 is primarily used from the IPP support Microsoft has provided for its Windows
65 (9X/ME/NT/2K/XP).
66 .RE

68 .sp
69 .ne 2
70 .na
71 \fBprint-uri\fR
72 .ad
73 .RS 26n
74 This is an optional IPP operation that allows client systems to submit a print
75 job with a reference (URL) for a single document. This operation is currently
76 not supported by the \fBmod_ipp\fR Apache Module.
77 .RE

79 .sp
80 .ne 2
81 .na
82 \fBvalidate-job\fR
83 .ad
84 .RS 26n
85 This is a required IPP operation that allows client systems to simulate the
86 submission of a print job to verify that the server is capable of handling the
87 job. This operation is supported by \fBmod_ipp\fR.
88 .RE

90 .sp
91 .ne 2
92 .na
93 \fBcreate-job\fR
94 .ad
95 .RS 26n
96 This is an optional IPP operation that allows client systems to submit a print
97 job. The operation is used with the \fBsend-document\fR and \fBsend-uri\fR
98 operations.
99 .RE

101 .sp
102 .ne 2
103 .na
104 \fBget-jobs\fR
105 .ad
106 .RS 26n
107 This is a required IPP operation that allows client systems to retrieve a list
108 of print jobs from the print service.
109 .RE

111 .sp
112 .ne 2
113 .na
114 \fBget-printer-attributes\fR
115 .ad
116 .RS 26n
117 This is a required IPP operation that allows client systems to retrieve
118 attributes from the print service that describes the named printer object.
119 .RE

121 .sp
122 .ne 2
123 .na
124 \fBpause-printer\fR
125 .ad
126 .RS 26n

```

```

127 This an optional IPP operation that allows client systems to stop job
128 processing on the named print queue.
129 .RE

131 .sp
132 .ne 2
133 .na
134 \fB\fBresume-printer\fR\fR
135 .ad
136 .RS 26n
137 This is an optional IPP operation that allows client systems to resume job
138 processing on the named print queue.
139 .RE

141 .sp
142 .ne 2
143 .na
144 \fB\fBpurge-jobs\fR\fR
145 .ad
146 .RS 26n
147 This is an optional IPP operation that allows client systems to cancel all
148 print jobs on the named print queue.
149 .RE

151 .sp
152 .ne 2
153 .na
154 \fB\fBsend-document\fR\fR
155 .ad
156 .RS 26n
157 This is a required IPP operation that allows client systems to add documents to
158 print jobs created with the \fBcreate-job\fR operation, but not yet submitted.
159 .RE

161 .sp
162 .ne 2
163 .na
164 \fB\fBsend-uri\fR\fR
165 .ad
166 .RS 26n
167 This is an optional IPP operation that allows a client system to add a document
168 reference (URI) to a print job created with the \fBcreate-job\fR operation, but
169 not yet submitted. This operation is currently not supported by the
170 \fBmod_ipp\fR Apache Module.
171 .RE

173 .sp
174 .ne 2
175 .na
176 \fB\fBcancel-job\fR\fR
177 .ad
178 .RS 26n
179 This is a required IPP operation that allows client systems to cancel print
180 jobs.
181 .RE

183 .sp
184 .ne 2
185 .na
186 \fB\fBget-job-attributes\fR\fR
187 .ad
188 .RS 26n
189 This is a required IPP operation that allows client systems to retrieve
190 attributes that describe a print job from the print service.
191 .RE

```

```

193 .sp
194 .ne 2
195 .na
196 \fB\fBhold-job\fR\fR
197 .ad
198 .RS 26n
199 This is an optional IPP operation that allows client systems to hold print
200 jobs.
201 .RE

203 .sp
204 .ne 2
205 .na
206 \fB\fBrelease-job\fR\fR
207 .ad
208 .RS 26n
209 This is an optional IPP operation that allows client systems to release print
210 jobs.
211 .RE

213 .sp
214 .ne 2
215 .na
216 \fB\fBrestart-job\fR\fR
217 .ad
218 .RS 26n
219 This is an optional IPP operation that allows client systems to restart print
220 jobs.
221 .RE

223 .sp
224 .ne 2
225 .na
226 \fB\fBall\fR\fR
227 .ad
228 .RS 26n
229 This is a place holder for enabling or disabling support for all IPP operations
230 implemented by the \fBmod_ipp\fR Apache module.
231 .RE

233 .sp
234 .ne 2
235 .na
236 \fB\fBrequired\fR\fR
237 .ad
238 .RS 26n
239 This is a place holder for enabling or disabling support for the required IPP
240 operations implemented by the \fBmod_ipp\fR Apache module.
241 .RE

243 .SH EXAMPLES
244 .LP
245 \fBExample 1 \fRUsing a Configuration File to Start a Standalone Apache Server
246 .sp
247 .LP
248 The following configuration file can be used to start a standalone Apache
249 server to respond to IPP request sent to port 631.

251 .sp
252 .in +2
253 .nf
254 ServerType standalone
255 ServerRoot "/usr/apache2/2.2"
255 ServerRoot "/usr/apache"
256 PidFile /var/run/httpd-standalone-ipp.pid
257 ErrorLog /var/log/logs/ipp-errors

```

```

259 Timeout 300
260 KeepAlive On
261 MaxKeepAliveRequests 100
262 KeepAliveTimeout 15

264 MinSpareServers 1
265 MaxSpareServers 3
266 StartServers 1
267 MaxClients 150
268 MaxRequestsPerChild 0

270 LoadModule ipp_module libexec/mod_ipp.so

272 ClearModuleList
273 AddModule mod_ipp.c
274 AddModule mod_so.c

276 Port 631

278 User lp
279 Group lp

281 ServerAdmin lp@localhost
282 DefaultType application/ipp

284 <IFModule mod_app>
285     <Location />
286         ipp-operation all on
287     </Location>
288 </IFModule mod_app>
289 .fi
290 .in -2

292 .sp
293 .LP
294 A more restrictive configuration might include the following parameters:

296 .sp
297 .in +2
298 .nf
299 <IFModule mod_app>
300     <Location />
301         ipp-operation all offn
302         ipp-operation required on
303     </Location>
304 </IFModule mod_app>
305 .fi
306 .in -2

308 .SH ATTRIBUTES
309 .sp
310 .LP
311 See \fBattributes\fR(5) or descriptions of the following attributes:
312 .sp

314 .sp
315 .TS
316 box;
317 c | c
318 l | l .
319 ATTRIBUTE TYPE ATTRIBUTE VALUE
320 -
321 Interface Stability External
322 .TE

```

```

324 .SH SEE ALSO
325 .sp
326 .LP
327 \fBman\fR(1), \fBcatman\fR(1M), \fBattributes\fR(5)
328 .sp
329 .LP
330 Herriot, R., Ed., Butler, S., Moore, P., Turner, R., Wenn, J. \fIRFC 2910,
331 Internet Printing Protocol/1.1: Encoding and Transport\fR. Network Working
332 Group. September 2000.
333 .sp
334 .LP
335 Hastings, T., Ed., Herriot, R., deBry, R., Isaacson, S., Powell, P. \fIRFC
336 2911, Internet Printing Protocol/1.1: Model and Semantics\fR. Network Working
337 Group. September 2000.
338 .sp
339 .LP
340 http://www.apache.org
341 .SH NOTES
342 .sp
343 .LP
344 Configuration file directives are processed in the order listed in the config
345 file. The default behavior is to enable support for all operations implemented
346 in the \fBmod_ipp\fR Apache module.
347 .sp
348 .LP
349 Since the Apache IPP listening service implements some capabilities that are
350 more of operator features, it may not be desirable to enable all IPP operations
351 without requiring user authentication on the Apache listening service.
352 .sp
353 .LP
354 The following is an example of a more reasonable configuration for Apache IPP
355 servers without user authentication enabled:
356 .sp
357 .in +2
358 .nf
359 ipp-operations all disabled
360 ipp-operations required enabled
361 .fi
362 .in -2

364 .sp
365 .LP
366 The printers and jobs available under this service can be accessed using URIs
367 of the following form:
368 .sp
369 .in +2
370 .nf
371 printer:
372     http://server\[:port\]/printers/{queue}
373     ipp://server\[:port\]/printers/{queue}
374 job:
375     http://server\[:port\]/printers/{queue}/{job-id}
376     ipp://server\[:port\]/printers/{queue}/{job-id}
377 .fi
378 .in -2

380 .sp
381 .LP
382 631 is the default IPP port and implied when the URI scheme is \fBipp\fR.
383 However, some client implementations do not recognize the \fBipp\fR URI scheme
384 and require \fBhttp://server:631/...fR instead. For example, Microsoft's IPP
385 client implementation does not recognize the \fBipp\fR scheme.
386 .sp
387 .LP
388 In addition to the documentation and man pages included with Solaris, more
389 information is available at http://www.apache.org

```

390 .sp  
391 .LP  
392 The \fBhttpd\fR(8) man page and other Apache man pages are provided with the  
393 programming modules. To view the Apache manual pages with the man command, add  
394 \fB/usr/apache2/2.2/man\fR to the \fBMANPATH\fR environment variable. See  
394 \fB/usr/apache/man\fR to the \fBMANPATH\fR environment variable. See  
395 \fBman\fR(1) for more information. Running \fBcatman\fR(1M) on the Apache  
396 manual pages is not supported.

\*\*\*\*\*

55288 Wed May 7 18:41:01 2014

new/usr/src/man/man5/filesystem.5

2837 - remove print/lp\* from gate and use CUPS from userland

\*\*\*\*\*

```

1  \" te
2  \" Copyright (c) 2008, Sun Microsystems, Inc. All Rights Reserved.
3  \" Copyright 1989 AT&T
4  \" The contents of this file are subject to the terms of the Common Development
5  \" See the License for the specific language governing permissions and limitat
6  \" fields enclosed by brackets \"[]\" replaced with your own identifying informat
7  .TH FILESYSTEM 5 \"Aug 26, 2013\"
8  .SH NAME
9  filesystem \- File system organization
10 .SH SYNOPSIS
11 .LP
12 .nf
13 /
14 .fi

16 .LP
17 .nf
18 /usr
19 .fi

21 .SH DESCRIPTION
22 .sp
23 .LP
24 The file system tree is organized for administrative convenience. Distinct
25 areas within the file system tree are provided for files that are private to
26 one machine, files that can be shared by multiple machines of a common
27 platform, files that can be shared by all machines, and home directories. This
28 organization allows sharable files to be stored on one machine but accessed by
29 many machines using a remote file access mechanism such as \fB NFS \fR. Grouping
30 together similar files makes the file system tree easier to upgrade and manage.
31 .sp
32 .LP
33 The file system tree consists of a root file system and a collection of
34 mountable file systems. The \fB mount \fR(2) program attaches mountable file
35 systems to the file system tree at mount points (directory entries) in the root
36 file system or other previously mounted file systems. Two file systems, \fB \fR
37 (the root) and \fB /usr \fR, must be mounted and \fB /var \fR must be accessible to
38 have a functional system. The root file system is mounted automatically by the
39 kernel at boot time; the \fB /usr \fR file system is mounted by the system
40 start-up script, which is run as part of the booting process. \fB /var \fR can be
41 mounted as its own file system or be part of \fB /usr \fR, as it is by default.
42 .sp
43 .LP
44 Certain locations, noted below, are approved installation locations for bundled
45 Foundation Solaris software. In some cases, the approved locations for bundled
46 software are also approved locations for add-on system software or for
47 applications. The following descriptions make clear where the two locations
48 differ. For example, \fB /etc \fR is the installation location for
49 platform-dependent configuration files that are bundled with Solaris software.
50 The analogous location for applications is \fB /etc/opt/\fR \fI package name \fR.
51 .sp
52 .LP
53 In the following descriptions, \fI subsystem \fR is a category of application or
54 system software, such as a window system (\fB dt \fR) or a language
55 (\fB javal.2 \fR)
56 .sp
57 .LP
58 The following descriptions make use of the terms \fB platform \fR,
59 \fB platform-dependent \fR, \fB platform-independent \fR, and
60 \fB platform-specific \fR. Platform refers to a machine's Instruction Set
61 Architecture or processor type, such as is returned by \fB uname \fR \fB -i \fR.

```

```

62 \fB Platform-dependent \fR refers to a file that is installed on all platforms
63 and whose contents vary depending on the platform. Like a platform-dependent
64 file, a \fB platform-independent \fR file is installed on all platforms. However,
65 the contents of the latter type remains the same on all platforms. An example
66 of a platform-dependent file is compiled, executable program. An example of a
67 platform-independent file is a standard configuration file, such as
68 \fB /etc/hosts \fR. Unlike a platform-dependent or a platform-independent file,
69 the \fB platform-specific \fR file is installed only on a subset of supported
70 platforms. Most platform-specific files are gathered under \fB /platform \fR and
71 \fB /usr/platform \fR.
72 .sp
73 .LP
74 In the following file or directory descriptions, GNOME stands for GNU Network
75 Object Model Environment. The GNOME Desktop is shipped with the Solaris
76 operating system.
77 .SS \"Root File System\"
78 .sp
79 .LP
80 The root file system contains files that are unique to each machine. It
81 contains the following directories:
82 .sp
83 .ne 2
84 .na
85 \fB \fB \fR
86 .ad
87 .sp .6
88 .RS 4n
89 Root of the overall file system name space.
90 .RE

92 .sp
93 .ne 2
94 .na
95 \fB \fB /dev \fR
96 .ad
97 .sp .6
98 .RS 4n
99 Primary location for special files. Typically, device files are built to match
100 the kernel and hardware configuration of the machine.
101 .RE

103 .sp
104 .ne 2
105 .na
106 \fB \fB /dev/cfg \fR
107 .ad
108 .sp .6
109 .RS 4n
110 Symbolic links to physical \fB ap_ids \fR
111 .RE

113 .sp
114 .ne 2
115 .na
116 \fB \fB /dev/cpu \fR
117 .ad
118 .sp .6
119 .RS 4n
120 Provides configuration and capability information about the processor type
121 .RE

123 .sp
124 .ne 2
125 .na
126 \fB \fB /dev/cua \fR
127 .ad

```

new/usr/src/man/man5/filesystem.5

128 .sp .6  
129 .RS 4n  
130 Device files for \fBuucp\fR.  
131 .RE  
  
133 .sp  
134 .ne 2  
135 .na  
136 \fB\fB/dev/dsk\fR\fR  
137 .ad  
138 .sp .6  
139 .RS 4n  
140 Block disk devices.  
141 .RE  
  
143 .sp  
144 .ne 2  
145 .na  
146 \fB\fB/dev/dtrace\fR\fR  
147 .ad  
148 .sp .6  
149 .RS 4n  
150 Pseudo-devices used by the DTrace framework.  
151 .RE  
  
153 .sp  
154 .ne 2  
155 .na  
156 \fB\fB/dev/dtrace/provider\fR\fR  
157 .ad  
158 .sp .6  
159 .RS 4n  
160 Pseudo-device drivers representing instrumentation providers for the DTrace  
161 framework.  
162 .RE  
  
164 .sp  
165 .ne 2  
166 .na  
167 \fB\fB/dev/fbs\fR\fR  
168 .ad  
169 .sp .6  
170 .RS 4n  
171 Frame buffer device files.  
172 .RE  
  
174 .sp  
175 .ne 2  
176 .na  
177 \fB\fB/dev/fd\fR\fR  
178 .ad  
179 .sp .6  
180 .RS 4n  
181 File descriptors.  
182 .RE  
  
184 .sp  
185 .ne 2  
186 .na  
187 \fB\fB/dev/md\fR\fR  
188 .ad  
189 .sp .6  
190 .RS 4n  
191 Logical volume management meta-disk devices.  
192 .RE

3

new/usr/src/man/man5/filesystem.5

194 .sp  
195 .ne 2  
196 .na  
197 \fB\fB/dev/net\fR\fR  
198 .ad  
199 .sp .6  
200 .RS 4n  
201 Network data-link interface devices.  
202 .RE  
  
204 .sp  
205 .ne 2  
206 .na  
207 \fB\fB/dev/printers\fR\fR  
208 .ad  
209 .sp .6  
210 .RS 4n  
211 USB printer device files.  
212 .RE  
  
214 .sp  
215 .ne 2  
216 .na  
217 \fB\fB/dev/pts\fR\fR  
218 .ad  
219 .sp .6  
220 .RS 4n  
221 Pseudo-terminal devices.  
222 .RE  
  
224 .sp  
225 .ne 2  
226 .na  
227 \fB\fB/dev/rdisk\fR\fR  
228 .ad  
229 .sp .6  
230 .RS 4n  
231 Raw disk devices.  
232 .RE  
  
234 .sp  
235 .ne 2  
236 .na  
237 \fB\fB/dev/rmt\fR\fR  
238 .ad  
239 .sp .6  
240 .RS 4n  
241 Raw tape devices.  
242 .RE  
  
244 .sp  
245 .ne 2  
246 .na  
247 \fB\fB/dev/sad\fR\fR  
248 .ad  
249 .sp .6  
250 .RS 4n  
251 Entry points for the \fBBSTREAMS\fR Administrative driver.  
252 .RE  
  
254 .sp  
255 .ne 2  
256 .na  
257 \fB\fB/dev/sound\fR\fR  
258 .ad  
259 .sp .6

4



```

260 .RS 4n
261 Audio device and audio device control files.
262 .RE

264 .sp
265 .ne 2
266 .na
267 \fb\fb/dev/swap\fR\fR
268 .ad
269 .sp .6
270 .RS 4n
271 Default swap device.
272 .RE

274 .sp
275 .ne 2
276 .na
277 \fb\fb/dev/term\fR\fR
278 .ad
279 .sp .6
280 .RS 4n
281 Terminal devices.
282 .RE

284 .sp
285 .ne 2
286 .na
287 \fb\fb/devices\fR\fR
288 .ad
289 .sp .6
290 .RS 4n
291 Physical device files.
292 .RE

294 .sp
295 .ne 2
296 .na
297 \fb\fb/etc\fR\fR
298 .ad
299 .sp .6
300 .RS 4n
301 Platform-dependent administrative and configuration files and databases that
302 are not shared among systems. \fb/etc\fR may be viewed as the directory that
303 defines the machine's identity. An approved installation location for bundled
304 Solaris software. The analogous location for add-on system software or for
305 applications is \fb/etc/opt/\fR\fIpackagename\fR.
306 .RE

308 .sp
309 .ne 2
310 .na
311 \fb\fb/etc/X11\fR\fR
312 .ad
313 .sp .6
314 .RS 4n
315 Xorg Xserver (X11) configuration files.
316 .RE

318 .sp
319 .ne 2
320 .na
321 \fb\fb/etc/acct\fR\fR
322 .ad
323 .sp .6
324 .RS 4n
325 Accounting system configuration information.

```

```

326 .RE

328 .sp
329 .ne 2
330 .na
331 \fb\fb/etc/apache2/2.2\fR\fR
332 .ad
333 .sp .6
334 .RS 4n
335 Apache configuration files.
336 .RE

338 .sp
339 .ne 2
340 .na
341 \fb\fb/etc/apoc\fR\fR
342 .ad
343 .sp .6
344 .RS 4n
345 Files for configuring Sun Java Desktop System Configuration Manager remote
346 access.
347 .RE

349 .sp
350 .ne 2
351 .na
352 \fb\fb/etc/bonobo-activation\fR\fR
353 .ad
354 .sp .6
355 .RS 4n
356 GNOME XML configuration file for identifying CORBA servers.
357 .RE

359 .sp
360 .ne 2
361 .na
362 \fb\fb/etc/cron.d\fR\fR
363 .ad
364 .sp .6
365 .RS 4n
366 Configuration information for \fbcron\fR(1M).
367 .RE

369 .sp
370 .ne 2
371 .na
372 \fb\fb/etc/dat\fR\fR
373 .ad
374 .sp .6
375 .RS 4n
376 Contains a list of interface adapters supported by uDAPL service providers.
377 .RE

379 .sp
380 .ne 2
381 .na
382 \fb\fb/etc/default\fR\fR
383 .ad
384 .sp .6
385 .RS 4n
386 Defaults information for various programs.
387 .RE

389 .sp
390 .ne 2

```

```

391 .na
392 \fB\fB/etc/devices\fR\fR
393 .ad
394 .sp .6
395 .RS 4n
396 Contains device-related data.
397 .RE

399 .sp
400 .ne 2
401 .na
402 \fB\fB/etc/dfs\fR\fR
403 .ad
404 .sp .6
405 .RS 4n
406 Configuration information for shared file systems.
407 .RE

409 .sp
410 .ne 2
411 .na
412 \fB\fB/etc/dhcp\fR\fR
413 .ad
414 .sp .6
415 .RS 4n
416 Dynamic Host Configuration Protocol (\fBDHCP\fR) configuration files.
417 .RE

419 .sp
420 .ne 2
421 .na
422 \fB\fB/etc/dmi\fR\fR
423 .ad
424 .sp .6
425 .RS 4n
426 Solstice Enterprise Agents configuration files.
427 .RE

429 .sp
430 .ne 2
431 .na
432 \fB\fB/etc/dt\fR\fR
433 .ad
434 .sp .6
435 .RS 4n
436 Desktop configuration files.
437 .RE

439 .sp
440 .ne 2
441 .na
442 \fB\fB/etc/flash\fR\fR
443 .ad
444 .sp .6
445 .RS 4n
446 Solaris Flash Archive configuration files.
447 .RE

449 .sp
450 .ne 2
451 .na
452 \fB\fB/etc/fm\fR\fR
453 .ad
454 .sp .6
455 .RS 4n
456 Fault manager configuration files. For more information, see \fBfmd\fR(1M).

```

```

457 .RE

459 .sp
460 .ne 2
461 .na
462 \fB\fB/etc/fonts\fR\fR
463 .ad
464 .sp .6
465 .RS 4n
466 Font configuration information.
467 .RE

469 .sp
470 .ne 2
471 .na
472 \fB\fB/etc/fs\fR\fR
473 .ad
474 .sp .6
475 .RS 4n
476 Binaries organized by file system types for operations required before
477 \fB/usr\fR is mounted.
478 .RE

480 .sp
481 .ne 2
482 .na
483 \fB\fB/etc/ftpd\fR\fR
484 .ad
485 .sp .6
486 .RS 4n
487 \fBftpd\fR configuration files.
488 .RE

490 .sp
491 .ne 2
492 .na
493 \fB\fB/etc/gconf\fR\fR
494 .ad
495 .sp .6
496 .RS 4n
497 GConf system configuration (including system defaults and system mandatory
498 settings)
499 .RE

501 .sp
502 .ne 2
503 .na
504 \fB\fB/etc/gimp\fR\fR
505 .ad
506 .sp .6
507 .RS 4n
508 GNU Image Manipulation Program (GIMP) configuration files.
509 .RE

511 .sp
512 .ne 2
513 .na
514 \fB\fB/etc/gnome\fR\fR
515 .ad
516 .sp .6
517 .RS 4n
518 GNOME Desktop configuration files.
519 .RE

521 .sp
522 .ne 2

```

```

523 .na
524 \fb\fb/etc/gnome-vfs-2.0\fr\fr
525 .ad
526 .sp .6
527 .RS 4n
528 Files for customizing GNOME 2.0 desktop menus.
529 .RE

531 .sp
532 .ne 2
533 .na
534 \fb\fb/etc/gnopernicus-1.0\fr\fr
535 .ad
536 .sp .6
537 .RS 4n
538 Configuration files for GNOME's Gnopernicus, an Assistive Technology (AT)
539 screen reader.
540 .RE

542 .sp
543 .ne 2
544 .na
545 \fb\fb/etc/gss\fr\fr
546 .ad
547 .sp .6
548 .RS 4n
549 Generic Security Service (\fbGSS\fr) Application Program Interface
550 configuration files.
551 .RE

553 .sp
554 .ne 2
555 .na
556 \fb\fb/etc/gtk\fr\fr
557 .ad
558 .sp .6
559 .RS 4n
560 GTK+ configuration files
561 .RE

563 .sp
564 .ne 2
565 .na
566 \fb\fb/etc/gtk-2.0\fr\fr
567 .ad
568 .sp .6
569 .RS 4n
570 GTK+ Pixbuf loaders and Input Method modules
571 .RE

573 .sp
574 .ne 2
575 .na
576 \fb\fb/etc/imq\fr\fr
577 .ad
578 .sp .6
579 .RS 4n
580 Sun Java System Message Queue security configuration files.
581 .RE

583 .sp
584 .ne 2
585 .na
586 \fb\fb/etc/inet\fr\fr
587 .ad
588 .sp .6

```

```

589 .RS 4n
590 Configuration files for Internet services.
591 .RE

593 .sp
594 .ne 2
595 .na
596 \fb\fb/etc/init.d\fr\fr
597 .ad
598 .sp .6
599 .RS 4n
600 Shell scripts for transitioning between run levels.
601 .RE

603 .sp
604 .ne 2
605 .na
606 \fb\fb/etc/krb5\fr\fr
607 .ad
608 .sp .6
609 .RS 4n
610 Kerberos configuration files.
611 .RE

613 .sp
614 .ne 2
615 .na
616 \fb\fb/etc/lib\fr\fr
617 .ad
618 .sp .6
619 .RS 4n
620 Shared libraries needed during booting.
621 .RE

623 .sp
624 .ne 2
625 .na
626 \fb\fb/etc/llc2\fr\fr
627 .ad
628 .sp .6
629 .RS 4n
630 Logical link control (\fbllc2\fr) driver configuration files.
631 .RE

633 .sp
634 .ne 2
635 .na
636 \fb\fb/etc/lp\fr\fr
637 .ad
638 .sp .6
639 .RS 4n
640 Configuration information for the printer subsystem.
641 .RE

643 .sp
644 .ne 2
645 .na
646 \fb\fb/etc/lvm\fr\fr
647 .ad
648 .sp .6
649 .RS 4n
650 Solaris Logical Volume Manager configuration files.
651 .RE

653 .sp
654 .ne 2

```

```

655 .na
656 \fB\fB/etc/mail\fR\fR
657 .ad
658 .sp .6
659 .RS 4n
660 Mail subsystem configuration.
661 .RE

663 .sp
664 .ne 2
665 .na
666 \fB\fB/etc/nca\fR\fR
667 .ad
668 .sp .6
669 .RS 4n
670 Solaris Network Cache and Accelerator (\fBNCA\fR) configuration files.
671 .RE

673 .sp
674 .ne 2
675 .na
676 \fB\fB/etc/net\fR\fR
677 .ad
678 .sp .6
679 .RS 4n
680 Configuration information for transport independent network services.
681 .RE

683 .sp
684 .ne 2
685 .na
686 \fB\fB/etc/nfs\fR\fR
687 .ad
688 .sp .6
689 .RS 4n
690 NFS server logging configuration file.
691 .RE

693 .sp
694 .ne 2
695 .na
696 \fB\fB/etc/opt\fR\fR
697 .ad
698 .sp .6
699 .RS 4n
700 Configuration information for optional packages.
701 .RE

703 .sp
704 .ne 2
705 .na
706 \fB\fB/etc/pango\fR\fR
707 .ad
708 .sp .6
709 .RS 4n
710 Pango configuration and module information.
711 .RE

713 .sp
714 .ne 2
715 .na
716 \fB\fB/etc/patch\fR\fR
717 .ad
718 .sp .6
719 .RS 4n
720 Configuration files for patch management.

```

```

721 .RE

723 .sp
724 .ne 2
725 .na
726 \fB\fB/etc/ppp\fR\fR
727 .ad
728 .sp .6
729 .RS 4n
730 Solaris \fBPPP\fR configuration files.
731 .RE

733 .sp
734 .ne 2
735 .na
736 \fB\fB/etc/rc0.d\fR\fR
737 .ad
738 .sp .6
739 .RS 4n
740 Scripts for entering or leaving run level 0. See \fBinit\fR(1M).
741 .RE

743 .sp
744 .ne 2
745 .na
746 \fB\fB/etc/rc1.d\fR\fR
747 .ad
748 .sp .6
749 .RS 4n
750 Scripts for entering or leaving run level 1. See \fBinit\fR(1M).
751 .RE

753 .sp
754 .ne 2
755 .na
756 \fB\fB/etc/rc2.d\fR\fR
757 .ad
758 .sp .6
759 .RS 4n
760 Scripts for entering or leaving run level 2. See \fBinit\fR(1M).
761 .RE

763 .sp
764 .ne 2
765 .na
766 \fB\fB/etc/rc3.d\fR\fR
767 .ad
768 .sp .6
769 .RS 4n
770 Scripts for entering or leaving run level 3. See \fBinit\fR(1M).
771 .RE

773 .sp
774 .ne 2
775 .na
776 \fB\fB/etc/rcS.d\fR\fR
777 .ad
778 .sp .6
779 .RS 4n
780 Scripts for bringing the system up in single user mode.
781 .RE

783 .sp
784 .ne 2
785 .na
786 \fB\fB/etc/rcm\fR\fR

```

```

787 .ad
788 .sp .6
789 .RS 4n
790 Directory for reconfiguration manager (RCM) custom scripts.
791 .RE

793 .sp
794 .ne 2
795 .na
796 \fb\fb/etc/rpcsec\fr\fr
797 .ad
798 .sp .6
799 .RS 4n
800 This directory might contain an \fBNIS+\fr authentication configuration file.
801 .RE

803 .sp
804 .ne 2
805 .na
806 \fb\fb/etc/saf\fr\fr
807 .ad
808 .sp .6
809 .RS 4n
810 Service Access Facility files.
811 .RE

813 .sp
814 .ne 2
815 .na
816 \fb\fb/etc/sasl\fr\fr
817 .ad
818 .sp .6
819 .RS 4n
820 Simple Authentication and Security Layer (SASL) server configuration files.
821 .RE

823 .sp
824 .ne 2
825 .na
826 \fb\fb/etc/security\fr\fr
827 .ad
828 .sp .6
829 .RS 4n
830 Solaris-delivered security configuration files (Audit, RBAC, crypto, Trusted
831 Extensions).
832 .RE

834 .sp
835 .ne 2
836 .na
837 \fb\fb/etc/sfw\fr\fr
838 .ad
839 .sp .6
840 .RS 4n
841 Platform-dependent administrative, configuration files and databases for
842 subsystems from \fb\usr/sfw\fr that are not shared among systems.
843 .RE

845 .sp
846 .ne 2
847 .na
848 \fb\fb/etc/sfw/samba\fr\fr
849 .ad
850 .sp .6
851 .RS 4n
852 Samba configuration files.

```

```

853 .RE

855 .sp
856 .ne 2
857 .na
858 \fb\fb/etc/skel\fr\fr
859 .ad
860 .sp .6
861 .RS 4n
862 Default profile scripts for new user accounts. See \fbUseradd\fr(1M).
863 .RE

865 .sp
866 .ne 2
867 .na
868 \fb\fb/etc/sma\fr\fr
869 .ad
870 .sp .6
871 .RS 4n
872 Systems Management Agent (SMA) configuration files.
873 .RE

875 .sp
876 .ne 2
877 .na
878 \fb\fb/etc/smartcard\fr\fr
879 .ad
880 .sp .6
881 .RS 4n
882 Solaris Smart Card configuration files.
883 .RE

885 .sp
886 .ne 2
887 .na
888 \fb\fb/etc/snmp\fr\fr
889 .ad
890 .sp .6
891 .RS 4n
892 Solstice Enterprise Agents configuration files.
893 .RE

895 .sp
896 .ne 2
897 .na
898 \fb\fb/etc/sound\fr\fr
899 .ad
900 .sp .6
901 .RS 4n
902 Sound Events configuration files.
903 .RE

905 .sp
906 .ne 2
907 .na
908 \fb\fb/etc/ssh\fr\fr
909 .ad
910 .sp .6
911 .RS 4n
912 Secure Shell configuration files. See \fbssh\fr(1)
913 .RE

915 .sp
916 .ne 2
917 .na
918 \fb\fb/etc/svc\fr\fr

```

```

919 .ad
920 .sp .6
921 .RS 4n
922 SMF service repository.
923 .RE

925 .sp
926 .ne 2
927 .na
928 \fB\fB/etc/sysevent\fR\fR
929 .ad
930 .sp .6
931 .RS 4n
932 \fBsyseventd\fR configuration files.
933 .RE

935 .sp
936 .ne 2
937 .na
938 \fB\fB/etc/\fIsubsystem\fR\fR
939 .ad
940 .sp .6
941 .RS 4n
942 Platform-dependent \fIsubsystem\fR configuration files that are not shared
943 among systems. An approved installation location for bundled Solaris software.
944 The analogous location for add-on system software or for applications is
945 \fB/etc/opt/\fR\fIpackage_name\fR.
946 .RE

948 .sp
949 .ne 2
950 .na
951 \fB\fB/etc/tm\fR\fR
952 .ad
953 .sp .6
954 .RS 4n
955 Trademark files; contents displayed at boot time.
956 .RE

958 .sp
959 .ne 2
960 .na
961 \fB\fB/etc/usb\fR\fR
962 .ad
963 .sp .6
964 .RS 4n
965 \fBUSB\fR configuration information.
966 .RE

968 .sp
969 .ne 2
970 .na
971 \fB\fB/etc/uucp\fR\fR
972 .ad
973 .sp .6
974 .RS 4n
975 \fBUUCP\fR configuration information. See \fBuucp\fR(1C).
976 .RE

978 .sp
979 .ne 2
980 .na
981 \fB\fB/etc/xml\fR\fR
982 .ad
983 .sp .6
984 .RS 4n

```

```

985 Extensible Markup Language (XML) catalog.
986 .RE

988 .sp
989 .ne 2
990 .na
991 \fB\fB/etc/zones\fR\fR
992 .ad
993 .sp .6
994 .RS 4n
995 Solaris Zones configuration files.
996 .RE

998 .sp
999 .ne 2
1000 .na
1001 \fB\fB/export\fR\fR
1002 .ad
1003 .sp .6
1004 .RS 4n
1005 Default root of the shared file system tree.
1006 .RE

1008 .sp
1009 .ne 2
1010 .na
1011 \fB\fB/home\fR\fR
1012 .ad
1013 .sp .6
1014 .RS 4n
1015 Default root of a subtree for user directories.
1016 .RE

1018 .sp
1019 .ne 2
1020 .na
1021 \fB\fB/kernel\fR\fR
1022 .ad
1023 .sp .6
1024 .RS 4n
1025 Subtree of platform-dependent loadable kernel modules required as part of the
1026 boot process. It includes the generic part of the core kernel that is
1027 platform-independent, \fB/kernel/genunix\fR. See \fBkernel\fR(1M) An approved
1028 installation location for bundled Solaris software and for add-on system
1029 software.
1030 .RE

1032 .sp
1033 .ne 2
1034 .na
1035 \fB\fB/kernel/drv\fR\fR
1036 .ad
1037 .sp .6
1038 .RS 4n
1039 32-bit x86 device drivers.
1040 .RE

1042 .sp
1043 .ne 2
1044 .na
1045 \fB\fB/kernel/drv/sparcv9\fR\fR
1046 .ad
1047 .sp .6
1048 .RS 4n
1049 64-bit \fBSPARC\fR device drivers.
1050 .RE

```

```

1052 .sp
1053 .ne 2
1054 .na
1055 \fb\fb/kernel/drv/amd64\fr\fr
1056 .ad
1057 .sp .6
1058 .RS 4n
1059 64-bit device drivers for 64-bit x86 platforms.
1060 .RE

1062 .sp
1063 .ne 2
1064 .na
1065 \fb\fb/kernel/dtrace\fr\fr
1066 .ad
1067 .sp .6
1068 .RS 4n
1069 Kernel modules representing components in the DTrace framework.
1070 .RE

1072 .sp
1073 .ne 2
1074 .na
1075 \fb\fb/kernel/genunix\fr\fr
1076 .ad
1077 .sp .6
1078 .RS 4n
1079 Platform-independent kernel.
1080 .RE

1082 .sp
1083 .ne 2
1084 .na
1085 \fb\fb/kernel/amd64/genunix\fr\fr
1086 .ad
1087 .sp .6
1088 .RS 4n
1089 64-bit, platform-independent kernel.
1090 .RE

1092 .sp
1093 .ne 2
1094 .na
1095 \fb\fb/kernel/\fI subsystem\fr/amd64\fr\fr
1096 .ad
1097 .sp .6
1098 .RS 4n
1099 64-bit x86 platform-dependent modules required for boot. An approved
1100 installation location for bundled Solaris software and for add-on system
1101 software.
1102 .RE

1104 .sp
1105 .ne 2
1106 .na
1107 \fb\fb/kernel/\fI subsystem\fr/sparcv9\fr\fr
1108 .ad
1109 .sp .6
1110 .RS 4n
1111 64-bit \fBSPARC\fr platform-dependent modules required for boot. An approved
1112 installation location for bundled Solaris software and for add-on system
1113 software.
1114 .RE

1116 .sp

```

```

1117 .ne 2
1118 .na
1119 \fb\fb/lib/svc/manifest\fr\fr
1120 .ad
1121 .sp .6
1122 .RS 4n
1123 SMF method scripts. An approved installation location for bundled Solaris
1124 software. The analogous location for add-on system software or for applications
1125 is \fb/opt/packageName/lib/svc/manifest\fr.
1126 .RE

1128 .sp
1129 .ne 2
1130 .na
1131 \fb\fb/mnt\fr\fr
1132 .ad
1133 .sp .6
1134 .RS 4n
1135 Default temporary mount point for file systems. This is an empty directory on
1136 which file systems can be temporarily mounted.
1137 .RE

1139 .sp
1140 .ne 2
1141 .na
1142 \fb\fb/net\fr\fr
1143 .ad
1144 .sp .6
1145 .RS 4n
1146 Temporary mount point for file systems that are mounted by the automounter.
1147 .RE

1149 .sp
1150 .ne 2
1151 .na
1152 \fb\fb/opt\fr\fr
1153 .ad
1154 .sp .6
1155 .RS 4n
1156 Subtree of a subtree for add-on application packages.
1157 .RE

1159 .sp
1160 .ne 2
1161 .na
1162 \fb\fb/platform\fr\fr
1163 .ad
1164 .sp .6
1165 .RS 4n
1166 Subtree of platform-specific objects which need to reside on the root
1167 filesystem. It contains a series of directories, one per supported platform.
1168 The semantics of the series of directories is equivalent to \fb\fr (root).
1169 .RE

1171 .sp
1172 .ne 2
1173 .na
1174 \fb\fb/platform/\fr\fb'\fr\fbuname\fr \fb-i\fr\fb\fr'\fr\fb/kernel\fr\fr
1175 .ad
1176 .sp .6
1177 .RS 4n
1178 Platform-specific modules required for boot. These modules have semantics
1179 equivalent to \fb/kernel\fr. It includes the file \fbunix\fr, the core kernel.
1180 See \fbkernel\fr(1M). An approved installation location for bundled Solaris
1181 software and for add-on system software.
1182 .RE

```

```

1184 .sp
1185 .ne 2
1186 .na
1187 \fB\fB/platform/\fR\fB'\fR\fBuname\fR \fB-m\fR\fB\fR\fB'\fR\fB/kernel\fR\fR
1188 .ad
1189 .sp .6
1190 .RS 4n
1191 Hardware class-specific modules required for boot. An approved installation
1192 location for bundled Solaris software and for add-on system software.
1193 .RE

1195 .sp
1196 .ne 2
1197 .na
1198 \fB\fB/platform/\fR\fB'\fR\fBuname\fR
1199 \fB-i\fR\fB\fR\fB'\fR\fB/kernel/\fR\fBisubsystem\fR/amd64\fR\fR
1200 .ad
1201 .sp .6
1202 .RS 4n
1203 x86 64-bit, platform-dependent modules required for boot. This is an approved
1204 installation location for bundled Solaris software.
1205 .RE

1207 .sp
1208 .ne 2
1209 .na
1210 \fB\fB/platform/\fR\fB'\fR\fBuname\fR
1211 \fB-i\fR\fB\fR\fB'\fR\fB/kernel/\fR\fBisubsystem\fR/sparcv9\fR\fR
1212 .ad
1213 .sp .6
1214 .RS 4n
1215 \fBSPARC\fR 64-bit platform-specific modules required for boot. An approved
1216 installation location for bundled Solaris software.
1217 .RE

1219 .sp
1220 .ne 2
1221 .na
1222 \fB\fB/platform/\fR\fB'\fR\fBuname\fR
1223 \fB-i\fR\fB\fR\fB'\fR\fB/kernel/sparcv9/unix\fR\fR
1224 .ad
1225 .sp .6
1226 .RS 4n
1227 64-bit platform-dependent kernel.
1228 .RE

1230 .sp
1231 .ne 2
1232 .na
1233 \fB\fB/platform/\fR\fB'\fR\fBuname\fR
1234 \fB-i\fR\fB\fR\fB'\fR\fB/kernel/unix\fR\fR
1235 .ad
1236 .sp .6
1237 .RS 4n
1238 32-bit platform-dependent kernel on i86 and a symlink to \fBsparcv9/unix\fR on
1239 SPARC.
1240 .RE

1242 .sp
1243 .ne 2
1244 .na
1245 \fB\fB/platform/\fR\fB'\fR\fBuname\fR \fB-i\fR\fB\fB\fR'\fR\fB/lib\fR\fR
1246 .ad
1247 .sp .6
1248 .RS 4n

```

```

1249 Platform-specific shared objects required for boot. Semantics are equivalent to
1250 \fB/lib\fR. An approved installation location for bundled Solaris software and
1251 for add-on system software.
1252 .RE

1254 .sp
1255 .ne 2
1256 .na
1257 \fB\fB/platform/\fR\fB'\fR\fBuname\fR \fB-i\fR\fB\fB\fR'\fR\fB/sbin\fR\fR
1258 .ad
1259 .sp .6
1260 .RS 4n
1261 Platform-specific administrative utilities required for boot. Semantics are
1262 equivalent to \fB/sbin\fR. An approved installation location for bundled
1263 Solaris software and for add-on system software.
1264 .RE

1266 .sp
1267 .ne 2
1268 .na
1269 \fB\fB/proc\fR\fR
1270 .ad
1271 .sp .6
1272 .RS 4n
1273 Root of a subtree for the process file system.
1274 .RE

1276 .sp
1277 .ne 2
1278 .na
1279 \fB\fB/sbin\fR\fR
1280 .ad
1281 .sp .6
1282 .RS 4n
1283 Essential executables used in the booting process and in manual system
1284 recovery. The full complement of utilities is available only after \fB/usr\fR
1285 is mounted. \fB/sbin\fR is an approved installation location for bundled
1286 Solaris software.
1287 .RE

1289 .sp
1290 .ne 2
1291 .na
1292 \fB\fB/system\fR\fR
1293 .ad
1294 .sp .6
1295 .RS 4n
1296 Mount point for the contract (CTFS) and object (OBJFS) file systems.
1297 .RE

1299 .sp
1300 .ne 2
1301 .na
1302 \fB\fB/tmp\fR\fR
1303 .ad
1304 .sp .6
1305 .RS 4n
1306 Temporary files; cleared during the boot operation.
1307 .RE

1309 .sp
1310 .ne 2
1311 .na
1312 \fB\fB/usr\fR\fR
1313 .ad
1314 .sp .6

```



```

1315 .RS 4n
1316 Mount point for the \fb\usr\fr file system. See description of \fb\usr\fr file
1317 system, below.
1318 .RE

1320 .sp
1321 .ne 2
1322 .na
1323 \fb\fb\var\fr\fr
1324 .ad
1325 .sp .6
1326 .RS 4n
1327 Root of a subtree for varying files. Varying files are files that are unique to
1328 a machine but that can grow to an arbitrary (that is, variable) size. An
1329 example is a log file. An approved installation location for bundled Solaris
1330 software. The analogous location for add-on system software or for applications
1331 is \fb\var/opt/\fr\fr\package\fr.
1332 .RE

1334 .sp
1335 .ne 2
1336 .na
1337 \fb\fb\var\adm\fr\fr
1338 .ad
1339 .sp .6
1340 .RS 4n
1341 System logging and accounting files.
1342 .RE

1344 .sp
1345 .ne 2
1346 .na
1347 \fb\fb\var\apache2\2.2\fr\fr
1348 .ad
1349 .sp .6
1350 .RS 4n
1351 Scripts, icons, logs, and cache pages for Apache web server.
1352 .RE

1354 .sp
1355 .ne 2
1356 .na
1357 \fb\fb\var\appserver\fr\fr
1358 .ad
1359 .sp .6
1360 .RS 4n
1361 Sun Java System Application Server administrative domain files.
1362 .RE

1364 .sp
1365 .ne 2
1366 .na
1367 \fb\fb\var\audit\fr\fr
1368 .ad
1369 .sp .6
1370 .RS 4n
1371 Default location for Solaris Audit log files.
1372 .RE

1374 .sp
1375 .ne 2
1376 .na
1377 \fb\fb\var\cores\fr\fr
1378 .ad
1379 .sp .6

```

```

1380 .RS 4n
1381 Directory provided for global core files storage.
1382 .RE

1384 .sp
1385 .ne 2
1386 .na
1387 \fb\fb\var\crash\fr\fr
1388 .ad
1389 .sp .6
1390 .RS 4n
1391 Default depository for kernel crash dumps.
1392 .RE

1394 .sp
1395 .ne 2
1396 .na
1397 \fb\fb\var\cron\fr\fr
1398 .ad
1399 .sp .6
1400 .RS 4n
1401 Log files for \fb\cron\fr(1M).
1402 .RE

1404 .sp
1405 .ne 2
1406 .na
1407 \fb\fb\var\dmide\fr\fr
1408 .ad
1409 .sp .6
1410 .RS 4n
1411 Solstice Enterprise Agents (\fb\SEA\fr) Desktop Management Interface (\fb\DMI\fr)
1412 run-time components.
1413 .RE

1415 .sp
1416 .ne 2
1417 .na
1418 \fb\fb\var\dt\fr\fr
1419 .ad
1420 .sp .6
1421 .RS 4n
1422 \fb\dtlogin\fr configuration files.
1423 .RE

1425 .sp
1426 .ne 2
1427 .na
1428 \fb\fb\var\fm\fr\fr
1429 .ad
1430 .sp .6
1431 .RS 4n
1432 Fault manager state files. For more information, see \fb\fm\fr(1M).
1433 .RE

1435 .sp
1436 .ne 2
1437 .na
1438 \fb\fb\var\img\fr\fr
1439 .ad
1440 .sp .6
1441 .RS 4n
1442 Message queue broker instance configuration file.
1443 .RE

1445 .sp

```

```

1446 .ne 2
1447 .na
1448 \fB\fB/var/ftp\fR\fR
1449 .ad
1450 .sp .6
1451 .RS 4n
1452 \fBFTP\fR server directory.
1453 .RE

1455 .sp
1456 .ne 2
1457 .na
1458 \fB\fB/var/inet\fR\fR
1459 .ad
1460 .sp .6
1461 .RS 4n
1462 IPv6 router state files.
1463 .RE

1465 .sp
1466 .ne 2
1467 .na
1468 \fB\fB/var/krb5\fR\fR
1469 .ad
1470 .sp .6
1471 .RS 4n
1472 Database and log files for Kerberos.
1473 .RE

1475 .sp
1476 .ne 2
1477 .na
1478 \fB\fB/var/ld\fR\fR
1479 .ad
1480 .sp .6
1481 .RS 4n
1482 Configuration files for runtime linker.
1483 .RE

1485 .sp
1486 .ne 2
1487 .na
1488 \fB\fB/var/ldap\fR\fR
1489 .ad
1490 .sp .6
1491 .RS 4n
1492 LDAP client configuration files.
1493 .RE

1495 .sp
1496 .ne 2
1497 .na
1498 \fB\fB/var/lib\fR\fR
1499 .ad
1500 .sp .6
1501 .RS 4n
1502 Directory for variable state information.
1503 .RE

1505 .sp
1506 .ne 2
1507 .na
1508 \fB\fB/var/log\fR\fR
1509 .ad
1510 .sp .6
1511 .RS 4n

```

```

1512 System log files.
1513 .RE

1515 .sp
1516 .ne 2
1517 .na
1518 \fB\fB/var/lp\fR\fR
1519 .ad
1520 .sp .6
1521 .RS 4n
1522 Line printer subsystem logging information.
1523 .RE

1525 .sp
1526 .ne 2
1527 .na
1528 \fB\fB/var/mail\fR\fR
1529 .ad
1530 .sp .6
1531 .RS 4n
1532 Directory where users' mail is kept.
1533 .RE

1535 .sp
1536 .ne 2
1537 .na
1538 \fB\fB/var/mysql\fR\fR
1539 .ad
1540 .sp .6
1541 .RS 4n
1542 Dynamic database directory for MySQL Database Management System.
1543 .RE

1545 .sp
1546 .ne 2
1547 .na
1548 \fB\fB/var/news\fR\fR
1549 .ad
1550 .sp .6
1551 .RS 4n
1552 Community service messages. This is not the same as USENET-style news.
1553 .RE

1555 .sp
1556 .ne 2
1557 .na
1558 \fB\fB/var/nfs\fR\fR
1559 .ad
1560 .sp .6
1561 .RS 4n
1562 NFS server log files.
1563 .RE

1565 .sp
1566 .ne 2
1567 .na
1568 \fB\fB/var/nis\fR\fR
1569 .ad
1570 .sp .6
1571 .RS 4n
1572 NIS+ databases.
1573 .RE

1575 .sp
1576 .ne 2
1577 .na

```

```

1578 \fB\fB/var/ntp\fR\fR
1579 .ad
1580 .sp .6
1581 .RS 4n
1582 Network Time Protocol (\fBNTP\fR) server state directory.
1583 .RE

1585 .sp
1586 .ne 2
1587 .na
1588 \fB\fB/var/opt\fR\fR
1589 .ad
1590 .sp .6
1591 .RS 4n
1592 Root of a subtree for varying files associated with optional software packages.
1593 An approved installation location for add-on system software and applications.
1594 .RE

1596 .sp
1597 .ne 2
1598 .na
1599 \fB\fB/var/patchsrv\fR\fR
1600 .ad
1601 .sp .6
1602 .RS 4n
1603 Patch management log files.
1604 .RE

1606 .sp
1607 .ne 2
1608 .na
1609 \fB\fB/var/preserve\fR\fR
1610 .ad
1611 .sp .6
1612 .RS 4n
1613 Backup files for \fBvi\fR(1) and \fBex\fR(1).
1614 .RE

1616 .sp
1617 .ne 2
1618 .na
1619 \fB\fB/var/run\fR\fR
1620 .ad
1621 .sp .6
1622 .RS 4n
1623 Temporary files which are not needed across reboots. Only root may modify the
1624 contents of this directory.
1625 .RE

1627 .sp
1628 .ne 2
1629 .na
1630 \fB\fB/var/sadm\fR\fR
1631 .ad
1632 .sp .6
1633 .RS 4n
1634 Databases maintained by the software package management utilities.
1635 .RE

1637 .sp
1638 .ne 2
1639 .na
1640 \fB\fB/var/sadm/system/logs\fR\fR
1641 .ad
1642 .sp .6
1643 .RS 4n

```

```

1644 Status log files produced by software management functions and/or applications.
1645 For example, log files produced for product installation. An approved
1646 installation location for bundled Solaris software and for add-on system
1647 software and applications.
1648 .RE

1650 .sp
1651 .ne 2
1652 .na
1653 \fB\fB/var/saf\fR\fR
1654 .ad
1655 .sp .6
1656 .RS 4n
1657 Service access facility logging and accounting files.
1658 .RE

1660 .sp
1661 .ne 2
1662 .na
1663 \fB\fB/var/samba\fR\fR
1664 .ad
1665 .sp .6
1666 .RS 4n
1667 Log and lock files for Samba.
1668 .RE

1670 .sp
1671 .ne 2
1672 .na
1673 \fB\fB/var/sma_snmp\fR\fR
1674 .ad
1675 .sp .6
1676 .RS 4n
1677 Systems Management Agent (SMA) security and MIB component information.
1678 .RE

1680 .sp
1681 .ne 2
1682 .na
1683 \fB\fB/var/snmp\fR\fR
1684 .ad
1685 .sp .6
1686 .RS 4n
1687 SNMP status and configuration information.
1688 .RE

1690 .sp
1691 .ne 2
1692 .na
1693 \fB\fB/var/spool\fR\fR
1694 .ad
1695 .sp .6
1696 .RS 4n
1697 Contains directories for files used in printer spooling, mail delivery,
1698 \fBcron\fR(1M), \fBbat\fR(1), and so forth.
1699 .RE

1701 .sp
1702 .ne 2
1703 .na
1704 \fB\fB/var/spool/clientmqueue\fR\fR
1705 .ad
1706 .sp .6
1707 .RS 4n
1708 \fBsendmail\fR(1M) client files.
1709 .RE

```

```

1711 .sp
1712 .ne 2
1713 .na
1714 \fB\fB/var/spool/cron\fR\fR
1715 .ad
1716 .sp .6
1717 .RS 4n
1718 \fB\fB/cron\fR(1M) and \fB\fB/Bat\fR(1) spooling files.
1719 .RE

1721 .sp
1722 .ne 2
1723 .na
1724 \fB\fB/var/spool/locks\fR\fR
1725 .ad
1726 .sp .6
1727 .RS 4n
1728 Spooling lock files.
1729 .RE

1731 .sp
1732 .ne 2
1733 .na
1734 \fB\fB/var/spool/lp\fR\fR
1735 .ad
1736 .sp .6
1737 .RS 4n
1738 Line printer spool files. See \fB\fB/lp\fR(1).
1739 .RE

1741 .sp
1742 .ne 2
1743 .na
1744 \fB\fB/var/spool/mqueue\fR\fR
1745 .ad
1746 .sp .6
1747 .RS 4n
1748 Mail queued for delivery.
1749 .RE

1751 .sp
1752 .ne 2
1753 .na
1754 \fB\fB/var/spool/pkg\fR\fR
1755 .ad
1756 .sp .6
1757 .RS 4n
1758 Spooled packages.
1759 .RE

1761 .sp
1762 .ne 2
1763 .na
1764 \fB\fB/var/spool/print\fR\fR
1765 .ad
1766 .sp .6
1767 .RS 4n
1768 LP print service client-side request staging area.
1769 .RE

1771 .sp
1772 .ne 2
1773 .na
1774 \fB\fB/var/spool/samba\fR\fR
1775 .ad

```

```

1776 .sp .6
1777 .RS 4n
1778 Samba print queue.
1779 .RE

1781 .sp
1782 .ne 2
1783 .na
1784 \fB\fB/var/spool/uucp\fR\fR
1785 .ad
1786 .sp .6
1787 .RS 4n
1788 Queued \fB\fB/uucp\fR(1C) jobs.
1789 .RE

1791 .sp
1792 .ne 2
1793 .na
1794 \fB\fB/var/spool/uucppublic\fR\fR
1795 .ad
1796 .sp .6
1797 .RS 4n
1798 Files deposited by \fB\fB/uucp\fR(1C).
1799 .RE

1801 .sp
1802 .ne 2
1803 .na
1804 \fB\fB/var/statmon\fR\fR
1805 .ad
1806 .sp .6
1807 .RS 4n
1808 Network status monitor files.
1809 .RE

1811 .sp
1812 .ne 2
1813 .na
1814 \fB\fB/var/svc/log\fR\fR
1815 .ad
1816 .sp .6
1817 .RS 4n
1818 SMF log files.
1819 .RE

1821 .sp
1822 .ne 2
1823 .na
1824 \fB\fB/var/svc/manifest\fR\fR
1825 .ad
1826 .sp .6
1827 .RS 4n
1828 SMF service manifests. An approved installation location for bundled, add-on
1829 system software and applications.
1830 .RE

1832 .sp
1833 .ne 2
1834 .na
1835 \fB\fB/var/svc/manifest/site\fR\fR
1836 .ad
1837 .sp .6
1838 .RS 4n
1839 Site-local SMF service manifests.
1840 .RE

```

```

1842 .sp
1843 .ne 2
1844 .na
1845 \fB\fB/var/tmp\fR\fR
1846 .ad
1847 .sp .6
1848 .RS 4n
1849 Files that vary in size or presence during normal system operations. This
1850 directory is \fBnot\fR cleared during the boot operation. An approved
1851 installation location for bundled Solaris software and for add-on system
1852 software and applications.
1853 .sp
1854 It is possible to change the default behavior for \fB/var/tmp\fR to clear all
1855 of the files except editor temporary files by setting the \fBclean_vartmp\fR
1856 property value of the \fBbrtmpfiles\fR service. This is done with the following
1857 commands:
1858 .sp
1859 .in +2
1860 .nf
1861 # \fBsvccfg -s svc:/system/rmtmpfiles setprop \e\fR
1862     \fBoptions/clean_vartmp = "true"\fR
1863 # \fBsvcadm refresh svc:/system/rmtmpfiles:default\fR
1864 .fi
1865 .in -2
1866 .sp

1868 The \fBsolaris.smf.value.rmtmpfiles\fR authorization is required to modify this
1869 property.
1870 .RE

1872 .sp
1873 .ne 2
1874 .na
1875 \fB\fB/var/uucp\fR\fR
1876 .ad
1877 .sp .6
1878 .RS 4n
1879 \fBUucp\fR(1C) log and status files.
1880 .RE

1882 .sp
1883 .ne 2
1884 .na
1885 \fB\fB/var/yp\fR\fR
1886 .ad
1887 .sp .6
1888 .RS 4n
1889 Databases needed for backwards compatibility with \fBNIS\fR and
1890 \fBypbind\fR(1M); unnecessary after full transition to \fBNIS+\fR.
1891 .RE

1893 .SS "\fB/usr\fR File System"
1894 .sp
1895 .LP
1896 Because it is desirable to keep the root file system small and not volatile, on
1897 disk-based systems larger file systems are often mounted on \fB/home\fR,
1898 \fB/opt\fR, \fB/usr\fR, and \fB/var\fR.
1899 .sp
1900 .LP
1901 The file system mounted on \fB/usr\fR contains platform-dependent and
1902 platform-independent sharable files. The subtree rooted at \fB/usr/share\fR
1903 contains platform-independent sharable files; the rest of the \fB/usr\fR tree
1904 contains platform-dependent files. By mounting a common remote file system, a
1905 group of machines with a common platform may share a single \fB/usr\fR file
1906 system. A single \fB/usr/share\fR file system can be shared by machines of any
1907 platform. A machine acting as a file server can share many different \fB/usr\fR

```

```

1908 file systems to support several different architectures and operating system
1909 releases. Clients usually mount \fB/usr\fR read-only so that they do not
1910 accidentally change any shared files.
1911 .sp
1912 .LP
1913 The \fB/usr\fR file system contains the following subdirectories:
1914 .sp
1915 .ne 2
1916 .na
1917 \fB\fB/usr/4lib\fR\fR
1918 .ad
1919 .sp .6
1920 .RS 4n
1921 \fBba.out\fR libraries for the Binary Compatibility Package.
1922 .RE

1924 .sp
1925 .ne 2
1926 .na
1927 \fB\fB/usr/5bin\fR\fR
1928 .ad
1929 .sp .6
1930 .RS 4n
1931 Symbolic link to the \fB/usr/bin\fR directory.
1932 .RE

1934 .sp
1935 .ne 2
1936 .na
1937 \fB\fB/usr/SUNwale\fR\fR
1938 .ad
1939 .sp .6
1940 .RS 4n
1941 Configuration files for Asian Lanuguage Environment (ALE).
1942 .RE

1944 .sp
1945 .ne 2
1946 .na
1947 \fB\fB/usr/X\fR\fR
1948 .ad
1949 .sp .6
1950 .RS 4n
1951 Symbolic link to the \fB/usr/openwin\fR directory.
1952 .RE

1954 .sp
1955 .ne 2
1956 .na
1957 \fB\fB/usr/X11\fR\fR
1958 .ad
1959 .sp .6
1960 .RS 4n
1961 Xorg Xserver (X11) executables and documentation.
1962 .RE

1964 .sp
1965 .ne 2
1966 .na
1967 \fB\fB/usr/adm\fR\fR
1968 .ad
1969 .sp .6
1970 .RS 4n
1971 Symbolic link to the \fB/var/adm\fR directory.
1972 .RE

```

```

1974 .sp
1975 .ne 2
1976 .na
1977 \fB\fB/usr/apache2/2.2\fR\fR
1978 \fB\fB/usr/apache\fR\fR
1979 .ad
1980 .sp .6
1981 .RS 4n
1982 Apache executables, loadable modules, and documentation.
1983 .RE

1984 .sp
1985 .ne 2
1986 .na
1987 \fB\fB/usr/appserver\fR\fR
1988 .ad
1989 .sp .6
1990 .RS 4n
1991 Sun Java System Application Server software.
1992 .RE

1994 .sp
1995 .ne 2
1996 .na
1997 \fB\fB/usr/benchmarks\fR\fR
1998 .ad
1999 .sp .6
2000 .RS 4n
2001 Directory for benchmarks.
2002 .RE

2004 .sp
2005 .ne 2
2006 .na
2007 \fB\fB/usr/bin\fR\fR
2008 .ad
2009 .sp .6
2010 .RS 4n
2011 Platform-dependent, user-invoked executables. These are commands users expect
2012 to be run as part of their normal \fB$PATH\fR. For executables that are
2013 different on a 64-bit system than on a 32-bit system, a wrapper that selects
2014 the appropriate executable is placed here. See \fBisaexec\fR(3C). An approved
2015 installation location for bundled Solaris software. The analogous location for
2016 add-on system software or for applications is \fB/opt/\fIpackagename\fR/bin\fR.
2017 .RE

2019 .sp
2020 .ne 2
2021 .na
2022 \fB\fB/usr/bin/amd64\fR\fR
2023 .ad
2024 .sp .6
2025 .RS 4n
2026 x86 64-bit, platform-dependent, user-invoked executables. This directory should
2027 not be part of a user's \fB$PATH\fR. A wrapper in \fB/usr/bin\fR should invoke
2028 the executable in this directory. See \fBisaexec\fR(3C). An approved
2029 installation location for bundled Solaris software. The analogous location for
2030 add-on system software or for applications is
2031 \fB/opt/\fIpackagename\fR/bin/amd64\fR.
2032 .RE

2034 .sp
2035 .ne 2
2036 .na
2037 \fB\fB/usr/bin/sparcv9\fR\fR
2038 .ad

```

```

2039 .sp .6
2040 .RS 4n
2041 \fB$PARC\fR platform-dependent, user-invoked executables. This directory should
2042 not be part of a user's \fB$PATH\fR. A wrapper in \fB/usr/bin\fR should invoke
2043 the executable in this directory. See \fBisaexec\fR(3C). An approved
2044 installation location for bundled Solaris software. The analogous location for
2045 add-on system software or for applications is
2046 \fB/opt/\fIpackagename\fR/bin/sparcv9\fR.
2047 .RE

2049 .sp
2050 .ne 2
2051 .na
2052 \fB\fB/usr/bin/\fI subsystem\fR\fR
2053 .ad
2054 .sp .6
2055 .RS 4n
2056 Platform-dependent user-invoked executables that are associated with
2057 \fI subsystem\fR. These are commands users expect to be run as part of their
2058 normal \fB$PATH\fR. An approved installation location for bundled Solaris
2059 software. The analogous location for add-on system software or for applications
2060 is \fB/opt/\fIpackagename\fR/bin\fR.
2061 .RE

2063 .sp
2064 .ne 2
2065 .na
2066 \fB\fB/usr/\fI subsystem\fR/bin\fR
2067 .ad
2068 .sp .6
2069 .RS 4n
2070 Platform-dependent user-invoked executables that are associated with
2071 \fI subsystem\fR. These are commands users expect to be run as part of their
2072 normal \fB$PATH\fR. An approved installation location for bundled Solaris
2073 software. The analogous location for add-on system software or for applications
2074 is \fB/opt/\fIpackagename\fR/bin\fR.
2075 .RE

2077 .sp
2078 .ne 2
2079 .na
2080 \fB\fB/usr/\fI subsystem\fR/bin/amd64\fR\fR
2081 .ad
2082 .sp .6
2083 .RS 4n
2084 x86 64-bit, platform-dependent, user-invoked executables. This directory should
2085 not be part of a user's \fB$PATH\fR. A wrapper in \fB/usr/bin\fR should invoke
2086 the executable in this directory. See \fBisaexec\fR(3C). An approved
2087 installation location for bundled Solaris software. The analogous location for
2088 add-on system software or for applications is
2089 \fB/opt/\fIpackagename\fR/bin/amd64\fR.
2090 .RE

2092 .sp
2093 .ne 2
2094 .na
2095 \fB\fB/usr/\fI subsystem\fR/bin/sparcv9\fR\fR
2096 .ad
2097 .sp .6
2098 .RS 4n
2099 \fB$PARC\fR 64-bit, platform-dependent, user-invoked executables. This
2100 directory should not be part of a user's \fB$PATH\fR. A wrapper in
2101 \fB/usr/bin\fR should invoke the executable in this directory. See
2102 \fBisaexec\fR(3C). An approved installation location for bundled Solaris
2103 software. The analogous location for add-on system software or for applications
2104 is \fB/opt/\fIpackagename\fR/bin/sparcv9\fR.

```

```

2105 .RE

2107 .sp
2108 .ne 2
2109 .na
2110 \fB\fB/usr/ccs\fR\fR
2111 .ad
2112 .sp .6
2113 .RS 4n
2114 C compilation system.
2115 .RE

2117 .sp
2118 .ne 2
2119 .na
2120 \fB\fB/usr/ccs/bin\fR\fR
2121 .ad
2122 .sp .6
2123 .RS 4n
2124 C compilation commands and system utilities.
2125 .RE

2127 .sp
2128 .ne 2
2129 .na
2130 \fB\fB/usr/ccs/lib\fR\fR
2131 .ad
2132 .sp .6
2133 .RS 4n
2134 Symbolic link to \fB/usr/lib\fR.
2135 .RE

2137 .sp
2138 .ne 2
2139 .na
2140 \fB\fB/usr/demo\fR\fR
2141 .ad
2142 .sp .6
2143 .RS 4n
2144 Demo programs and data.
2145 .RE

2147 .sp
2148 .ne 2
2149 .na
2150 \fB\fB/usr/dict\fR\fR
2151 .ad
2152 .sp .6
2153 .RS 4n
2154 Symbolic link to the \fB/usr/share/lib/dict\fR directory, which contains the
2155 dictionary file used by the \fBUNIX\fR spell program.
2156 .RE

2158 .sp
2159 .ne 2
2160 .na
2161 \fB\fB/usr/dt\fR\fR
2162 .ad
2163 .sp .6
2164 .RS 4n
2165 root of a subtree for CDE software.
2166 .RE

2168 .sp
2169 .ne 2
2170 .na

```

```

2171 \fB\fB/usr/dt/bin\fR\fR
2172 .ad
2173 .sp .6
2174 .RS 4n
2175 Primary location for CDE system utilities.
2176 .RE

2178 .sp
2179 .ne 2
2180 .na
2181 \fB\fB/usr/dt/include\fR\fR
2182 .ad
2183 .sp .6
2184 .RS 4n
2185 Header files for CDE software.
2186 .RE

2188 .sp
2189 .ne 2
2190 .na
2191 \fB\fB/usr/dt/lib\fR\fR
2192 .ad
2193 .sp .6
2194 .RS 4n
2195 Libraries for CDE software.
2196 .RE

2198 .sp
2199 .ne 2
2200 .na
2201 \fB\fB/usr/dt/share/man\fR\fR
2202 .ad
2203 .sp .6
2204 .RS 4n
2205 On-line reference manual pages for CDE software.
2206 .RE

2208 .sp
2209 .ne 2
2210 .na
2211 \fB\fB/usr/games\fR\fR
2212 .ad
2213 .sp .6
2214 .RS 4n
2215 An empty directory, a remnant of the SunOS 4.0/4.1 software.
2216 .RE

2218 .sp
2219 .ne 2
2220 .na
2221 \fB\fB/usr/gnome\fR\fR
2222 .ad
2223 .sp .6
2224 .RS 4n
2225 This is an obsolete directory where 3rd party programs can install their
2226 applications and \fBpixmap\fR files. It is supported for backwards
2227 compatibility.
2228 .RE

2230 .sp
2231 .ne 2
2232 .na
2233 \fB\fB/usr/include\fR\fR
2234 .ad
2235 .sp .6
2236 .RS 4n

```

```

2237 Include headers (for C programs).
2238 .RE

2240 .sp
2241 .ne 2
2242 .na
2243 \fB\fB/usr/j2se\fR\fR
2244 .ad
2245 .sp .6
2246 .RS 4n
2247 Java 2 SDK executables, loadable modules, and documentation.
2248 .RE

2250 .sp
2251 .ne 2
2252 .na
2253 \fB\fB/usr/java\fI*\fR\fR\fR
2254 .ad
2255 .sp .6
2256 .RS 4n
2257 Directories containing Java programs and libraries.
2258 .RE

2260 .sp
2261 .ne 2
2262 .na
2263 \fB\fB/usr/jdk\fI*\fR\fR\fR
2264 .ad
2265 .sp .6
2266 .RS 4n
2267 Java Platform virtual machine and core class libraries.
2268 .RE

2270 .sp
2271 .ne 2
2272 .na
2273 \fB\fB/usr/kernel\fR\fR
2274 .ad
2275 .sp .6
2276 .RS 4n
2277 Subtree of platform-dependent loadable kernel modules, not needed in the root
2278 filesystem. An approved installation location for bundled Solaris software.
2279 .RE

2281 .sp
2282 .ne 2
2283 .na
2284 \fB\fB/usr/kvm\fR\fR
2285 .ad
2286 .sp .6
2287 .RS 4n
2288 A mount point, retained for backward compatibility, that formerly contained
2289 platform-specific binaries and libraries.
2290 .RE

2292 .sp
2293 .ne 2
2294 .na
2295 \fB\fB/usr/lib\fR\fR
2296 .ad
2297 .sp .6
2298 .RS 4n
2299 Platform-dependent libraries, various databases, commands and daemons not
2300 invoked directly by a human user. An approved installation location for bundled
2301 Solaris software. The analogous location for add-on system software or for
2302 applications is \fB/opt/\fIpackage\fR/lib\fR.

```

```

2303 .RE

2305 .sp
2306 .ne 2
2307 .na
2308 \fB\fB/usr/lib/32\fR\fR
2309 .ad
2310 .sp .6
2311 .RS 4n
2312 Symbolic link to \fB/usr/lib\fR.
2313 .RE

2315 .sp
2316 .ne 2
2317 .na
2318 \fB\fB/usr/lib/64\fR\fR
2319 .ad
2320 .sp .6
2321 .RS 4n
2322 Symbolic link to the most portable 64-bit Solaris interfaces, on both SPARC and
2323 x86 platforms.
2324 .RE

2326 .sp
2327 .ne 2
2328 .na
2329 \fB\fB/usr/lib/acct\fR\fR
2330 .ad
2331 .sp .6
2332 .RS 4n
2333 Accounting scripts and binaries. See \fBacct\fR(1M).
2334 .RE

2336 .sp
2337 .ne 2
2338 .na
2339 \fB\fB/usr/lib/adb\fR\fR
2340 .ad
2341 .sp .6
2342 .RS 4n
2343 \fB\badb\fR accounting scripts.
2344 .RE

2346 .sp
2347 .ne 2
2348 .na
2349 \fB\fB/usr/lib/amd64\fR\fR
2350 .ad
2351 .sp .6
2352 .RS 4n
2353 Platform-dependent libraries, various databases, commands and daemons not
2354 invoked directly by a human user on 64-bit x86. An approved installation
2355 location for bundled Solaris software. The analogous location for add-on system
2356 software or for applications is \fB/opt/\fIpackage\fR/lib/amd64\fR.
2357 .RE

2359 .sp
2360 .ne 2
2361 .na
2362 \fB\fB/usr/lib/autofs\fR\fR
2363 .ad
2364 .sp .6
2365 .RS 4n
2366 Contains the \fB\bautomountd\fR executable.
2367 .RE

```



```

2369 .sp
2370 .ne 2
2371 .na
2372 \fB\fB/usr/lib/cfgadm\fR\fR
2373 .ad
2374 .sp .6
2375 .RS 4n
2376 Contains \fBcfgadm\fR hardware-specific driver plugins.
2377 .RE

2379 .sp
2380 .ne 2
2381 .na
2382 \fB\fB/usr/lib/class\fR\fR
2383 .ad
2384 .sp .6
2385 .RS 4n
2386 Scheduling-class-specific directories containing executables for
2387 \fBprioctl\fR(1) and \fBdispadm\fR(1M).
2388 .RE

2390 .sp
2391 .ne 2
2392 .na
2393 \fB\fB/usr/lib/crypto\fR\fR
2394 .ad
2395 .sp .6
2396 .RS 4n
2397 Contains the kernel-level cryptographic framework daemon (\fBkcfid\fR).
2398 .RE

2400 .sp
2401 .ne 2
2402 .na
2403 \fB\fB/usr/lib/devfsadm\fR\fR
2404 .ad
2405 .sp .6
2406 .RS 4n
2407 Contains \fBdevfsadm\fR, the daemon version of \fBdevfsadm\fR.
2408 .RE

2410 .sp
2411 .ne 2
2412 .na
2413 \fB\fB/usr/lib/dict\fR\fR
2414 .ad
2415 .sp .6
2416 .RS 4n
2417 Database files for \fBspell\fR(1).
2418 .RE

2420 .sp
2421 .ne 2
2422 .na
2423 \fB\fB/usr/lib/dns\fR\fR
2424 .ad
2425 .sp .6
2426 .RS 4n
2427 Contains DNS resolver libraries.
2428 .RE

2430 .sp
2431 .ne 2
2432 .na
2433 \fB\fB/usr/lib/dtrace\fR\fR
2434 .ad

```

```

2435 .sp .6
2436 .RS 4n
2437 Contains \fBdtrace\fR D source files.
2438 .RE

2440 .sp
2441 .ne 2
2442 .na
2443 \fB\fB/usr/lib/flash\fR\fR
2444 .ad
2445 .sp .6
2446 .RS 4n
2447 Contains Solaris flash archive deployment scripts.
2448 .RE

2450 .sp
2451 .ne 2
2452 .na
2453 \fB\fB/usr/lib/fm\fR\fR
2454 .ad
2455 .sp .6
2456 .RS 4n
2457 Contains \fBfmd\fR, the fault manager daemon and the fault manager library.
2458 .RE

2460 .sp
2461 .ne 2
2462 .na
2463 \fB\fB/usr/lib/font\fR\fR
2464 .ad
2465 .sp .6
2466 .RS 4n
2467 \fBftroff\fR(1) font description files.
2468 .RE

2470 .sp
2471 .ne 2
2472 .na
2473 \fB\fB/usr/lib/fs\fR\fR
2474 .ad
2475 .sp .6
2476 .RS 4n
2477 File system type dependent modules; generally not intended to be invoked
2478 directly by the user.
2479 .RE

2481 .sp
2482 .ne 2
2483 .na
2484 \fB\fB/usr/lib/gss\fR\fR
2485 .ad
2486 .sp .6
2487 .RS 4n
2488 Secure services-related libraries.
2489 .RE

2491 .sp
2492 .ne 2
2493 .na
2494 \fB\fB/usr/lib/iconv\fR\fR
2495 .ad
2496 .sp .6
2497 .RS 4n
2498 Conversion tables for \fBiconv\fR(1).
2499 .RE

```

```

2501 .sp
2502 .ne 2
2503 .na
2504 \fB\fB/usr/lib/inet\fR\fR
2505 .ad
2506 .sp .6
2507 .RS 4n
2508 Contains many network-related daemons and libraries.
2509 .RE

2511 .sp
2512 .ne 2
2513 .na
2514 \fB\fB/usr/lib/ipf\fR\fR
2515 .ad
2516 .sp .6
2517 .RS 4n
2518 Contains \fBIPFILTER.LICENCE\fR and \fBipftest\fR.
2519 .RE

2521 .sp
2522 .ne 2
2523 .na
2524 \fB\fB/usr/lib/ipqosconf\fR\fR
2525 .ad
2526 .sp .6
2527 .RS 4n
2528 IPQoS configuration utility.
2529 .RE

2531 .sp
2532 .ne 2
2533 .na
2534 \fB\fB/usr/lib/krb5\fR\fR
2535 .ad
2536 .sp .6
2537 .RS 4n
2538 Contains the Kerberos database propagation program and libraries.
2539 .RE

2541 .sp
2542 .ne 2
2543 .na
2544 \fB\fB/usr/lib/ld\fR\fR
2545 .ad
2546 .sp .6
2547 .RS 4n
2548 Contains the map files for the \fBld\fR link editor.
2549 .RE

2551 .sp
2552 .ne 2
2553 .na
2554 \fB\fB/usr/lib/ldap\fR\fR
2555 .ad
2556 .sp .6
2557 .RS 4n
2558 Contains LDAP client configuration utilities.
2559 .RE

2561 .sp
2562 .ne 2
2563 .na
2564 \fB\fB/usr/lib/libp\fR\fR
2565 .ad
2566 .sp .6

```

```

2567 .RS 4n
2568 Profiled libraries.
2569 .RE

2571 .sp
2572 .ne 2
2573 .na
2574 \fB\fB/usr/lib/llc2\fR\fR
2575 .ad
2576 .sp .6
2577 .RS 4n
2578 Contains logical link control (\fBllc2\fR) driver configuration files.
2579 .RE

2581 .sp
2582 .ne 2
2583 .na
2584 \fB\fB/usr/lib/locale\fR\fR
2585 .ad
2586 .sp .6
2587 .RS 4n
2588 Localization databases.
2589 .RE

2591 .sp
2592 .ne 2
2593 .na
2594 \fB\fB/usr/lib/lp\fR\fR
2595 .ad
2596 .sp .6
2597 .RS 4n
2598 Line printer subsystem databases and back-end executables.
2599 .RE

2601 .sp
2602 .ne 2
2603 .na
2604 \fB\fB/usr/lib/netsh\fR\fR
2605 .ad
2606 .sp .6
2607 .RS 4n
2608 Internet network services.
2609 .RE

2611 .sp
2612 .ne 2
2613 .na
2614 \fB\fB/usr/lib/nfs\fR\fR
2615 .ad
2616 .sp .6
2617 .RS 4n
2618 Auxiliary NFS-related programs and daemons.
2619 .RE

2621 .sp
2622 .ne 2
2623 .na
2624 \fB\fB/usr/lib/nis\fR\fR
2625 .ad
2626 .sp .6
2627 .RS 4n
2628 Contains NIS+ administrative commands.
2629 .RE

2631 .sp
2632 .ne 2

```

```

2633 .na
2634 \fB\fB/usr/lib/picl\fR\fR
2635 .ad
2636 .sp .6
2637 .RS 4n
2638 Platform Information and Control Library.
2639 .RE

2641 .sp
2642 .ne 2
2643 .na
2644 \fB\fB/usr/lib/pool\fR\fR
2645 .ad
2646 .sp .6
2647 .RS 4n
2648 Contains the automated resource pools partitioning daemon (\fBpooled\fR) and
2649 associated libraries.
2650 .RE

2652 .sp
2653 .ne 2
2654 .na
2655 \fB\fB/usr/lib/power\fR\fR
2656 .ad
2657 .sp .6
2658 .RS 4n
2659 Power management daemon, \fBpowerd\fR.
2660 .RE

2662 .sp
2663 .ne 2
2664 .na
2665 \fB\fB/usr/lib/print\fR\fR
2666 .ad
2667 .sp .6
2668 .RS 4n
2669 Contains \fB\lp\fR conversion scripts and the \fB\lpd\fR daemon.
2670 .RE

2672 .sp
2673 .ne 2
2674 .na
2675 \fB\fB/usr/lib/rcap\fR\fR
2676 .ad
2677 .sp .6
2678 .RS 4n
2679 Resource cap enforcement daemon, \fBrcapd\fR.
2680 .RE

2682 .sp
2683 .ne 2
2684 .na
2685 \fB\fB/usr/lib/rcm\fR\fR
2686 .ad
2687 .sp .6
2688 .RS 4n
2689 Contains the Reconfiguration and Coordination Manager daemon (\fBrcm_daemon\fR)
2690 and RCM scripts.
2691 .RE

2693 .sp
2694 .ne 2
2695 .na
2696 \fB\fB/usr/lib/refer\fR\fR
2697 .ad
2698 .sp .6

```

```

2699 .RS 4n
2700 Auxiliary programs for \fBprefer\fR(1).
2701 .RE

2703 .sp
2704 .ne 2
2705 .na
2706 \fB\fB/usr/lib/rmmount\fR\fR
2707 .ad
2708 .sp .6
2709 .RS 4n
2710 Removable media mounter shared objects.
2711 .RE

2713 .sp
2714 .ne 2
2715 .na
2716 \fB\fB/usr/lib/sa\fR\fR
2717 .ad
2718 .sp .6
2719 .RS 4n
2720 Scripts and commands for the system activity report package. See \fB\bsar\fR(1).
2721 .RE

2723 .sp
2724 .ne 2
2725 .na
2726 \fB\fB/usr/lib/saf\fR\fR
2727 .ad
2728 .sp .6
2729 .RS 4n
2730 Auxiliary programs and daemons related to the service access facility.
2731 .RE

2733 .sp
2734 .ne 2
2735 .na
2736 \fB\fB/usr/lib/sasl\fR\fR
2737 .ad
2738 .sp .6
2739 .RS 4n
2740 Simple Authentication and Security Layer (SASL) plug-in modules.
2741 .RE

2743 .sp
2744 .ne 2
2745 .na
2746 \fB\fB/usr/lib/secure\fR\fR
2747 .ad
2748 .sp .6
2749 .RS 4n
2750 Default trusted libraries.
2751 .RE

2753 .sp
2754 .ne 2
2755 .na
2756 \fB\fB/usr/lib/security\fR\fR
2757 .ad
2758 .sp .6
2759 .RS 4n
2760 Solaris security plug-in modules.
2761 .RE

2763 .sp
2764 .ne 2

```

```

2765 .na
2766 \fB\fB/usr/lib/smartcard\fR\fR
2767 .ad
2768 .sp .6
2769 .RS 4n
2770 IFD handler libraries.
2771 .RE

2773 .sp
2774 .ne 2
2775 .na
2776 \fB\fB/usr/lib/smedia\fR\fR
2777 .ad
2778 .sp .6
2779 .RS 4n
2780 Removable media device server daemon, \fB\rpc.smsserverd\fR.
2781 .RE

2783 .sp
2784 .ne 2
2785 .na
2786 \fB\fB/usr/lib/sparcv9\fR\fR
2787 .ad
2788 .sp .6
2789 .RS 4n
2790 \fBSPARC\fR 64-bit, platform-dependent libraries, various databases, commands
2791 and daemons not invoked directly by a human user. An approved installation
2792 location for bundled Solaris software. The analogous location for add-on system
2793 software or for applications is \fB/opt/\fIpackagename\fR/lib/sparcv9\fR.
2794 .RE

2796 .sp
2797 .ne 2
2798 .na
2799 \fB\fB/usr/lib/spell\fR\fR
2800 .ad
2801 .sp .6
2802 .RS 4n
2803 Auxiliary programs and databases for \fBspell\fR(1). This directory is only
2804 present when the Binary Compatibility Package is installed.
2805 .RE

2807 .sp
2808 .ne 2
2809 .na
2810 \fB\fB/usr/lib/ssh\fR\fR
2811 .ad
2812 .sp .6
2813 .RS 4n
2814 Contains the Secure Shell daemon (\fBsshd\fR) and supporting programs.
2815 .RE

2817 .sp
2818 .ne 2
2819 .na
2820 \fB\fB/usr/lib/\fIsubsystem\fR\fR
2821 .ad
2822 .sp .6
2823 .RS 4n
2824 Platform-dependent libraries, various databases, commands and daemons that are
2825 associated with \fIsubsystem\fR and that are not invoked directly by a human
2826 user. An approved installation location for bundled Solaris software. The
2827 analogous location for add-on system software or for applications is
2828 \fB/opt/\fIpackagename\fR/lib\fR.
2829 .RE

```

```

2831 .sp
2832 .ne 2
2833 .na
2834 \fB\fB/usr/lib/\fIsubsystem\fR/amd64\fR\fR
2835 .ad
2836 .sp .6
2837 .RS 4n
2838 x86 64-bit, platform-dependent libraries, various databases, commands and
2839 daemons that are associated with \fIsubsystem\fR and that are not invoked
2840 directly by a human user. An approved installation location for bundled Solaris
2841 software. The analogous location for add-on system software or for applications
2842 is \fB/opt/\fIpackagename\fR/lib/amd64\fR.
2843 .RE

2845 .sp
2846 .ne 2
2847 .na
2848 \fB\fB/usr/lib/\fIsubsystem\fR/sparcv9\fR\fR
2849 .ad
2850 .sp .6
2851 .RS 4n
2852 \fBSPARC\fR 64-bit, platform-dependent libraries, various databases, commands
2853 and daemons that are associated with \fIsubsystem\fR and that are not invoked
2854 directly by a human user. An approved installation location for bundled Solaris
2855 software. The analogous location for add-on system software or for applications
2856 is \fB/opt/\fIpackagename\fR/lib/sparcv9\fR.
2857 .RE

2859 .sp
2860 .ne 2
2861 .na
2862 \fB\fB/usr/\fIsubsystem\fR/lib\fR\fR
2863 .ad
2864 .sp .6
2865 .RS 4n
2866 Platform-dependent libraries, various databases, commands and daemons not
2867 invoked directly by a human user. An approved installation location for bundled
2868 Solaris software. The analogous location for add-on system software or for
2869 applications is \fB/opt/\fIpackagename\fR/lib\fR.
2870 .RE

2872 .sp
2873 .ne 2
2874 .na
2875 \fB\fB/usr/\fIsubsystem\fR/lib/amd64\fR\fR
2876 .ad
2877 .sp .6
2878 .RS 4n
2879 x86 64-bit, platform-dependent libraries, various databases, commands and
2880 daemons that are associated with \fIsubsystem\fR and that are not invoked
2881 directly by a human user. An approved installation location for bundled Solaris
2882 software. The analogous location for add-on system software or for applications
2883 is \fB/opt/\fIpackagename\fR/lib/amd64\fR.
2884 .RE

2886 .sp
2887 .ne 2
2888 .na
2889 \fB\fB/usr/\fIsubsystem\fR/lib/sparcv9\fR\fR
2890 .ad
2891 .sp .6
2892 .RS 4n
2893 \fBSPARC\fR 64-bit, platform-dependent libraries, various databases, commands
2894 and daemons that are associated with \fIsubsystem\fR and that are not invoked
2895 directly by a human user. An approved installation location for bundled Solaris
2896 software. The analogous location for add-on system software or for applications

```

2897 is \fB/opt/\fIpackagename\fR/lib/sparcv9\fR.  
 2898 .RE

2900 .sp  
 2901 .ne 2  
 2902 .na  
 2903 \fB\fB/usr/lib/sysevent\fR\fR  
 2904 .ad  
 2905 .sp .6  
 2906 .RS 4n  
 2907 Contains the system event notification daemon (\fBsyseventd\fR) and the  
 2908 \fBsyseventd\fR loadable module (SLM) repository.  
 2909 .RE

2911 .sp  
 2912 .ne 2  
 2913 .na  
 2914 \fB\fB/usr/lib/uucp\fR\fR  
 2915 .ad  
 2916 .sp .6  
 2917 .RS 4n  
 2918 Auxiliary programs and daemons for \fBuucp\fR(1C).  
 2919 .RE

2921 .sp  
 2922 .ne 2  
 2923 .na  
 2924 \fB\fB/usr/lib/webconsole\fR\fR  
 2925 .ad  
 2926 .sp .6  
 2927 .RS 4n  
 2928 Sun Java web console programs and scripts.  
 2929 .RE

2931 .sp  
 2932 .ne 2  
 2933 .na  
 2934 \fB\fB/usr/lib/zones\fR\fR  
 2935 .ad  
 2936 .sp .6  
 2937 .RS 4n  
 2938 Zone administration daemon (\fBzoneamd\fR).  
 2939 .RE

2941 .sp  
 2942 .ne 2  
 2943 .na  
 2944 \fB\fB/usr/local\fR\fR  
 2945 .ad  
 2946 .sp .6  
 2947 .RS 4n  
 2948 Not part of the SVR4-based Solaris distribution. The \fB/usr\fR directory is  
 2949 exclusively for software bundled with the Solaris operating system. If needed  
 2950 for storing machine-local add-on software, create the directory  
 2951 \fB/opt/local\fR and make \fB/usr/local\fR a symbolic link to \fB/opt/local\fR.  
 2952 The \fB/opt\fR directory or filesystem is for storing add-on software to the  
 2953 system.  
 2954 .RE

2956 .sp  
 2957 .ne 2  
 2958 .na  
 2959 \fB\fB/usr/mail\fR\fR  
 2960 .ad  
 2961 .sp .6  
 2962 .RS 4n

2963 Symbolic link to the \fB/var/mail\fR directory.  
 2964 .RE

2966 .sp  
 2967 .ne 2  
 2968 .na  
 2969 \fB\fB/usr/man\fR\fR  
 2970 .ad  
 2971 .sp .6  
 2972 .RS 4n  
 2973 Symbolic link to the \fB/usr/share/man\fR directory.  
 2974 .RE

2976 .sp  
 2977 .ne 2  
 2978 .na  
 2979 \fB\fB/usr/net/servers\fR\fR  
 2980 .ad  
 2981 .sp .6  
 2982 .RS 4n  
 2983 Entry points for foreign name service requests relayed using the network  
 2984 listener. See \fBlisten\fR(1M).  
 2985 .RE

2987 .sp  
 2988 .ne 2  
 2989 .na  
 2990 \fB\fB/usr/news\fR\fR  
 2991 .ad  
 2992 .sp .6  
 2993 .RS 4n  
 2994 Symbolic link to the \fB/var/news\fR directory.  
 2995 .RE

2997 .sp  
 2998 .ne 2  
 2999 .na  
 3000 \fB\fB/usr/old\fR\fR  
 3001 .ad  
 3002 .sp .6  
 3003 .RS 4n  
 3004 Programs that are being phased out.  
 3005 .RE

3007 .sp  
 3008 .ne 2  
 3009 .na  
 3010 \fB\fB/usr/openwin\fR\fR  
 3011 .ad  
 3012 .sp .6  
 3013 .RS 4n  
 3014 Installation or mount point for the OpenWindows software.  
 3015 .RE

3017 .sp  
 3018 .ne 2  
 3019 .na  
 3020 \fB\fB/usr/perl5\fR\fR  
 3021 .ad  
 3022 .sp .6  
 3023 .RS 4n  
 3024 Perl 5 programs and documentation  
 3025 .RE

3027 .sp  
 3028 .ne 2

```

3029 .na
3030 \fB\fB/usr/platform\fR\fR
3031 .ad
3032 .sp .6
3033 .RS 4n
3034 Subtree of platform-specific objects which does not need to reside on the root
3035 filesystem. It contains a series of directories, one per supported platform.
3036 The semantics of the series of directories is equivalent to \fB/platform\fR,
3037 except for subdirectories which do not provide utility under one or the other
3038 (for example, \fB/platform/include\fR is not needed).
3039 .RE

3041 .sp
3042 .ne 2
3043 .na
3044 \fB\fB/usr/platform/\fR\fB'\fR\fBuname\fR
3045 \fB-i\fR\fB\fR\fB'\fR\fB/include\fR\fR
3046 .ad
3047 .sp .6
3048 .RS 4n
3049 Symbolic link to \fB/./\fR\fB'\fR\fBuname\fR
3050 \fB-i\fR\fB\fR\fB'\fR\fB/include\fR. Platform-specific system (\fBsys\fR,
3051 \fBvm\fR) header files with semantics equivalent to \fB/usr/include\fR. An
3052 approved installation location for bundled Solaris software and for add-on
3053 system software.
3054 .RE

3056 .sp
3057 .ne 2
3058 .na
3059 \fB\fB/usr/platform/\fR\fB'\fR\fBuname\fR \fB-i\fR\fB\fR\fB'\fR\fB/lib\fR\fR
3060 .ad
3061 .sp .6
3062 .RS 4n
3063 Platform-specific shared objects with semantics equivalent to \fB/usr/lib\fR.
3064 An approved installation location for bundled Solaris software and for add-on
3065 system software.
3066 .RE

3068 .sp
3069 .ne 2
3070 .na
3071 \fB\fB/usr/platform/\fR\fB'\fR\fBuname\fR
3072 \fB-i\fR\fB\fR\fB'\fR\fB/lib/\fR\fBisubsystem\fR/amd64\fR\fR
3073 .ad
3074 .sp .6
3075 .RS 4n
3076 x86 64-bit, platform-specific daemon and shared objects. An approved
3077 installation location for bundled Solaris software and for add-on system
3078 software.
3079 .RE

3081 .sp
3082 .ne 2
3083 .na
3084 \fB\fB/usr/platform/\fR\fB'\fR\fBuname\fR \fB-i\fR\fB\fR\fB'\fR\fB/sbin\fR\fR
3085 .ad
3086 .sp .6
3087 .RS 4n
3088 Platform-specific system administration utilities with semantics equivalent to
3089 \fB/usr/sbin\fR. An approved installation location for bundled Solaris software
3090 and for add-on system software.
3091 .RE

3093 .sp
3094 .ne 2

```

```

3095 .na
3096 \fB\fB/usr/preserve\fR\fR
3097 .ad
3098 .sp .6
3099 .RS 4n
3100 Symbolic link to the \fB/var/preserve\fR directory.
3101 .RE

3103 .sp
3104 .ne 2
3105 .na
3106 \fB\fB/usr/proc\fR\fR
3107 .ad
3108 .sp .6
3109 .RS 4n
3110 Directory for the \fBproc\fR tools.
3111 .RE

3113 .sp
3114 .ne 2
3115 .na
3116 \fB\fB/usr/pub\fR\fR
3117 .ad
3118 .sp .6
3119 .RS 4n
3120 Symbolic link to \fB/share/lib/pub\fR, which contains files for online man page
3121 and character processing.
3122 .RE

3124 .sp
3125 .ne 2
3126 .na
3127 \fB\fB/usr/sadm\fR\fR
3128 .ad
3129 .sp .6
3130 .RS 4n
3131 System administration files and directories.
3132 .RE

3134 .sp
3135 .ne 2
3136 .na
3137 \fB\fB/usr/sadm/bin\fR\fR
3138 .ad
3139 .sp .6
3140 .RS 4n
3141 Binaries for the Form and Menu Language Interpreter (\fBFMLI\fR) scripts. See
3142 \fBfmli(1)\fR.
3143 .RE

3145 .sp
3146 .ne 2
3147 .na
3148 \fB\fB/usr/sadm/install\fR\fR
3149 .ad
3150 .sp .6
3151 .RS 4n
3152 Executables and scripts for package management.
3153 .RE

3155 .sp
3156 .ne 2
3157 .na
3158 \fB\fB/usr/sbin\fR\fR
3159 .ad
3160 .sp .6

```

```

3161 .RS 4n
3162 Platform-dependent executables for system administration, expected to be run
3163 only by system administrators. An approved installation location for bundled
3164 Solaris software. The analogous location for add-on system software or for
3165 applications is \fB/opt/\fIpackagename\fR/sbin\fR.
3166 .RE

3168 .sp
3169 .ne 2
3170 .na
3171 \fB\fB/usr/sbin/install.d\fR\fR
3172 .ad
3173 .sp .6
3174 .RS 4n
3175 Custom Jumpstart scripts and executables.
3176 .RE

3178 .sp
3179 .ne 2
3180 .na
3181 \fB\fB/usr/sbin/sparc7\fR and \fB\sparc9\fR\fR
3182 .ad
3183 .sp .6
3184 .RS 4n
3185 32-bit and 64-bit versions of commands.
3186 .RE

3188 .sp
3189 .ne 2
3190 .na
3191 \fB\fB/usr/sbin/amd64\fR\fR
3192 .ad
3193 .sp .6
3194 .RS 4n
3195 64-bit x86versions of commands.
3196 .RE

3198 .sp
3199 .ne 2
3200 .na
3201 \fB\fB/usr/sbin/\fI subsystem\fR\fR
3202 .ad
3203 .sp .6
3204 .RS 4n
3205 Platform-dependent executables for system administration, expected to be run
3206 only by system administrators, and associated with \fI subsystem\fR. An approved
3207 installation location for bundled Solaris software. The analogous location for
3208 add-on system software or for applications is
3209 \fB/opt/\fIpackagename\fR/sbin\fR.
3210 .RE

3212 .sp
3213 .ne 2
3214 .na
3215 \fB\fB/usr/\fI subsystem\fR/sbin\fR\fR
3216 .ad
3217 .sp .6
3218 .RS 4n
3219 Platform-dependent executables for system administration, expected to be run
3220 only by system administrators, and associated with \fI subsystem\fR. An approved
3221 installation location for bundled Solaris software. The analogous location for
3222 add-on system software or for applications is
3223 \fB/opt/\fIpackagename\fR/sbin\fR.
3224 .RE

3226 .sp

```

```

3227 .ne 2
3228 .na
3229 \fB\fB/usr/sfw\fR\fR
3230 .ad
3231 .sp .6
3232 .RS 4n
3233 \fBGNU\fR and open source executables, libraries, and documentation.
3234 .RE

3236 .sp
3237 .ne 2
3238 .na
3239 \fB\fB/usr/share\fR\fR
3240 .ad
3241 .sp .6
3242 .RS 4n
3243 Platform-independent sharable files. An approved installation location for
3244 bundled Solaris software.
3245 .RE

3247 .sp
3248 .ne 2
3249 .na
3250 \fB\fB/usr/share/aclocal\fR\fR
3251 .ad
3252 .sp .6
3253 .RS 4n
3254 Open source \fBm4\fR files.
3255 .RE

3257 .sp
3258 .ne 2
3259 .na
3260 \fB\fB/usr/share/applications\fR\fR
3261 .ad
3262 .sp .6
3263 .RS 4n
3264 Open source desktop applications files.
3265 .RE

3267 .sp
3268 .ne 2
3269 .na
3270 \fB\fB/usr/share/audio\fR\fR
3271 .ad
3272 .sp .6
3273 .RS 4n
3274 Sample audio files.
3275 .RE

3277 .sp
3278 .ne 2
3279 .na
3280 \fB\fB/usr/share/glib-2.0\fR\fR
3281 .ad
3282 .sp .6
3283 .RS 4n
3284 Makefile for \fBglib\fR.
3285 .RE

3287 .sp
3288 .ne 2
3289 .na
3290 \fB\fB/usr/share/gnome\fR\fR
3291 .ad
3292 .sp .6

```

```

3293 .RS 4n
3294 GNOME desktop and application data.
3295 .RE

3297 .sp
3298 .ne 2
3299 .na
3300 \fB\fB/usr/share/gtk-2.0\fR\fR
3301 .ad
3302 .sp .6
3303 .RS 4n
3304 GTK+ application data and demos
3305 .RE

3307 .sp
3308 .ne 2
3309 .na
3310 \fB\fB/usr/share/gtk-doc\fR\fR
3311 .ad
3312 .sp .6
3313 .RS 4n
3314 API documentation for libraries which use gtk-doc documentation format, which
3315 mostly includes desktop interfaces.
3316 .RE

3318 .sp
3319 .ne 2
3320 .na
3321 \fB\fB/usr/share/icons\fR\fR
3322 .ad
3323 .sp .6
3324 .RS 4n
3325 Sun Java Desktop icons.
3326 .RE

3328 .sp
3329 .ne 2
3330 .na
3331 \fB\fB/usr/share/idl\fR\fR
3332 .ad
3333 .sp .6
3334 .RS 4n
3335 Open source Interface Definition Language (IDL) files.
3336 .RE

3338 .sp
3339 .ne 2
3340 .na
3341 \fB\fB/usr/share/intltool\fR\fR
3342 .ad
3343 .sp .6
3344 .RS 4n
3345 XML translation tools.
3346 .RE

3348 .sp
3349 .ne 2
3350 .na
3351 \fB\fB/usr/share/ipfilter\fR\fR
3352 .ad
3353 .sp .6
3354 .RS 4n
3355 Open source IP Filter sample files.
3356 .RE

3358 .sp

```

```

3359 .ne 2
3360 .na
3361 \fB\fB/usr/share/javadoc\fR\fR
3362 .ad
3363 .sp .6
3364 .RS 4n
3365 Help files for Message Queue broker and Smart Card applications.
3366 .RE

3368 .sp
3369 .ne 2
3370 .na
3371 \fB\fB/usr/share/lib\fR\fR
3372 .ad
3373 .sp .6
3374 .RS 4n
3375 Platform-independent sharable databases. An approved installation location for
3376 bundled Solaris software.
3377 .RE

3379 .sp
3380 .ne 2
3381 .na
3382 \fB\fB/usr/share/lib/dict\fR\fR
3383 .ad
3384 .sp .6
3385 .RS 4n
3386 Contains word list for \fBspell\fR(1).
3387 .RE

3389 .sp
3390 .ne 2
3391 .na
3392 \fB\fB/usr/share/lib/keytables\fR\fR
3393 .ad
3394 .sp .6
3395 .RS 4n
3396 Keyboard layout description tables.
3397 .RE

3399 .sp
3400 .ne 2
3401 .na
3402 \fB\fB/usr/share/lib/mailx\fR\fR
3403 .ad
3404 .sp .6
3405 .RS 4n
3406 Help files for \fBmailx\fR(1).
3407 .RE

3409 .sp
3410 .ne 2
3411 .na
3412 \fB\fB/usr/share/lib/nterm\fR\fR
3413 .ad
3414 .sp .6
3415 .RS 4n
3416 \fBnroff\fR(1) terminal tables.
3417 .RE

3419 .sp
3420 .ne 2
3421 .na
3422 \fB\fB/usr/share/lib/pub\fR\fR
3423 .ad
3424 .sp .6

```



```

3425 .RS 4n
3426 Character set data files.
3427 .RE

3429 .sp
3430 .ne 2
3431 .na
3432 \fB\fB/usr/share/lib/tabset\fR\fR
3433 .ad
3434 .sp .6
3435 .RS 4n
3436 Tab setting escape sequences.
3437 .RE

3439 .sp
3440 .ne 2
3441 .na
3442 \fB\fB/usr/share/lib/terminfo\fR\fR
3443 .ad
3444 .sp .6
3445 .RS 4n
3446 Terminal description files for \fBterminfo\fR(4).
3447 .RE

3449 .sp
3450 .ne 2
3451 .na
3452 \fB\fB/usr/share/lib/tmac\fR\fR
3453 .ad
3454 .sp .6
3455 .RS 4n
3456 Macro packages and related files for text processing tools, for example,
3457 \fBnroff\fR(1) and \fBtroff\fR(1).
3458 .RE

3460 .sp
3461 .ne 2
3462 .na
3463 \fB\fB/usr/share/lib/zoneinfo\fR\fR
3464 .ad
3465 .sp .6
3466 .RS 4n
3467 Time zone information.
3468 .RE

3470 .sp
3471 .ne 2
3472 .na
3473 \fB\fB/usr/share/man\fR\fR
3474 .ad
3475 .sp .6
3476 .RS 4n
3477 Platform-independent sharable manual pages. An approved installation location
3478 for bundled Solaris software. The analogous location for add-on system software
3479 or for applications is \fB/opt/\fIpackage\fRname\fR/man\fR.
3480 .RE

3482 .sp
3483 .ne 2
3484 .na
3485 \fB\fB/usr/share/omf\fR\fR
3486 .ad
3487 .sp .6
3488 .RS 4n
3489 GNOME Scrollkeeper database files.
3490 .RE

```

```

3492 .sp
3493 .ne 2
3494 .na
3495 \fB\fB/usr/share/pixmaps\fR\fR
3496 .ad
3497 .sp .6
3498 .RS 4n
3499 Sun Java graphics.
3500 .RE

3502 .sp
3503 .ne 2
3504 .na
3505 \fB\fB/usr/share/scrollkeeper\fR\fR
3506 .ad
3507 .sp .6
3508 .RS 4n
3509 GNOME Scrollkeeper templates and \fBxslt\fR files.
3510 .RE

3512 .sp
3513 .ne 2
3514 .na
3515 \fB\fB/usr/share/sgml\fR\fR
3516 .ad
3517 .sp .6
3518 .RS 4n
3519 Open source SGML files.
3520 .RE

3522 .sp
3523 .ne 2
3524 .na
3525 \fB\fB/usr/share/sounds\fR\fR
3526 .ad
3527 .sp .6
3528 .RS 4n
3529 Sound files.
3530 .RE

3532 .sp
3533 .ne 2
3534 .na
3535 \fB\fB/usr/share/src\fR\fR
3536 .ad
3537 .sp .6
3538 .RS 4n
3539 Source code for kernel, utilities, and libraries.
3540 .RE

3542 .sp
3543 .ne 2
3544 .na
3545 \fB\fB/usr/share/themes\fR\fR
3546 .ad
3547 .sp .6
3548 .RS 4n
3549 GNOME 2.0 Desktop themes.
3550 .RE

3552 .sp
3553 .ne 2
3554 .na
3555 \fB\fB/usr/share/webconsole\fR\fR
3556 .ad

```



new/usr/src/pkg/manifests/developer-opensolaris-osnet.mf

1

```
*****
3187 Wed May 7 18:41:03 2014
new/usr/src/pkg/manifests/developer-opensolaris-osnet.mf
2837 - remove print/lp* from gnu and use CUPS from userland
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #

22 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.

24 set name=pkg.fmri value=pkg:/developer/opensolaris/osnet@${PKGVERS}
25 set name=pkg.description \
26     value="Dependencies required to build the OS/Net Consolidation."
27 set name=pkg.summary \
28     value="Dependencies required to build the OS/Net Consolidation."
29 set name=info.classification \
30     value="org.opensolaris.category.2008:Development/Distribution Tools"
31 set name=org.opensolaris.noincorp value=true
32 set name=variant.arch value=${ARCH}
33 depend fmri=data/docbook@0.5.11-0.133 type=require
34 depend fmri=developer/astdev@0.5.11-0.133 type=require
35 depend fmri=developer/build/make@0.5.11-0.133 type=require
36 depend fmri=developer/build/onbld@0.5.11-0.133 type=require
37 ${i386_ONLY}depend fmri=developer/gnu-binutils@2.19-0.133 type=require
38 depend fmri=developer/java/jdk@0.5.11-0.133 type=require
39 depend fmri=developer/lexer/flex@2.5.35-0.133 type=require
40 depend fmri=developer/object-file@0.5.11-0.133 type=require
41 depend fmri=developer/parser/bison@2.3-0.133 type=require
42 depend fmri=developer/versioning/mercurial@1.3.1-0.133 type=require
43 depend fmri=library/glib2@0.5.11-0.133 type=require
44 depend fmri=library/libxml2@2.7.6-0.133 type=require
45 depend fmri=library/libxslt@0.5.11-0.133 type=require
46 depend fmri=library/nspr/header-nspr@0.5.11-0.133 type=require
47 depend fmri=library/perl-5/xml-parser@0.5.11-0.133 type=require
48 depend fmri=library/security/trousers@0.3.2-0.133 type=require
49 depend fmri=print/cups@1.4.2-0.133 type=require
50 depend fmri=print/filter/ghostscript@8.64-0.133 type=require
51 depend fmri=runtime/perl-510/extra@5.10.0-0.133 type=require
52 depend fmri=runtime/perl-510@5.10.0-0.133 type=require
53 depend fmri=system/header@0.5.11-0.133 type=require
54 depend fmri=system/library/dbus@0.5.11-0.133 type=require
55 depend fmri=system/library/install@0.5.11-0.133 type=require
56 depend fmri=system/library/libdbus-glib@0.5.11-0.133 type=require
57 depend fmri=system/library/libdbus@0.5.11-0.133 type=require
58 depend fmri=system/library/math/header-math@0.5.11-0.133 type=require
59 depend fmri=system/library/mozilla-nss/header-nss@0.5.11-0.133 type=require
60 depend fmri=system/management/product-registry@0.5.11-0.133 type=require
61 depend fmri=system/management/snmp/net-snmp@5.4.1-0.133 type=require
```

new/usr/src/pkg/manifests/developer-opensolaris-osnet.mf

2

```
62 depend fmri=text/gnu-gettext@0.16.1-0.133 type=require
63 depend fmri=web/server/apache-22@2.2.26-0.133 type=require
63 depend fmri=web/server/apache-13@1.3.41-0.133 type=require
```

new/usr/src/pkg/manifests/print-lp-ipp-ipp-listener.mf

1

```
*****
2482 Wed May 7 18:41:07 2014
new/usr/src/pkg/manifests/print-lp-ipp-ipp-listener.mf
2837 - remove print/lp* from gate and use CUPS from userland
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 #
26 set name=pkg.fmri value=pkg:/print/lp/ipp/ipp-listener@$(PKGVERS)
27 set name=pkg.description \
28     value="Internet Printing Protocol(IPP) Apache module for listening service"
29 set name=pkg.summary value="Internet Printing Protocol(IPP) listener module"
30 set name=info.classification \
31     value=org.opensolaris.category.2008:System/Printing
32 set name=variant.arch value=$(ARCH)
33 dir path=etc group=sys
34 dir path=etc/apache2
35 dir path=etc/apache2/2.2
36 dir path=etc/apache
37 dir path=lib
38 dir path=lib/svc
39 dir path=lib/svc/manifest group=sys
40 dir path=lib/svc/manifest/application group=sys
41 dir path=lib/svc/manifest/application/print group=sys
42 dir path=usr group=sys
43 dir path=usr/apache2
44 dir path=usr/apache2/2.2
45 dir path=usr/apache2/2.2/libexec
46 dir path=usr/apache
47 dir path=usr/apache/libexec
48 dir path=var group=sys
49 dir path=var/lp group=lp mode=0775 owner=lp
50 dir path=var/lp/ipp-listener
51 file path=etc/apache2/2.2/httpd-standalone-ipp.conf \
52     file path=etc/apache/httpd-standalone-ipp.conf \
53     original_name=SUNWippl:etc/apache/httpd-standalone-ipp.conf preserve=true
54 file path=lib/svc/manifest/application/print/ipp-listener.xml group=sys \
55     mode=0444
56 file path=usr/apache2/2.2/libexec/mod_ipp.so mode=0555
57 file path=usr/apache/libexec/mod_ipp.so mode=0555
58 file path=var/lp/ipp-listener/index.html mode=0444
59 legacy pkg=SUNWipplr \
60     desc="Internet Printing Protocol(IPP) Apache configuration for service modul
61     name="Internet Printing Protocol(IPP) listener, (root)" \
```

new/usr/src/pkg/manifests/print-lp-ipp-ipp-listener.mf

2

```
57     version=13.1,REV=2009.11.11
58 legacy pkg=SUNWipplu \
59     desc="Internet Printing Protocol(IPP) Apache module for listening service" \
60     name="Internet Printing Protocol(IPP) listener module (/usr)" \
61     version=13.1,REV=2009.11.11
62 license cr_Sun license=cr_Sun
63 license lic_CDDL license=lic_CDDL
```