

## new/exception\_lists/check\_rtime

1

```
*****
9421 Wed Aug 8 20:59:25 2012
new/exception_lists/check_rtime
Bump Apache dependency to Apache 2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #

22 #
23 # Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights reserved.
24 #

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

42 # Directory hierarchies to skip completely
43 SKIP      ^usr/lib/libc/          # optimized libc
44 SKIP      ^usr/lib/rcm/          # 4426119
45 SKIP      ^usr/per15/           # alan's taking care of these :-(
46 SKIP      ^usr/src/              # no objects in source code

48 # Individual files that we don't examine
49 SKIP      ^boot/grub/bin/grub$   # Apache loadable module
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 SKIP      ^usr/lib/fps/sun4u/UltraSPARC.*/fptest$ # USIII specific extns. cause ldd noise on USIII bld. m/c
53 SKIP      ^usr/MACH(lib)/lddstub$ # lddstub has no dependencies
54 SKIP      ^usr/MACH(lib)/libssagent\.so\.1$ # 4328854
55 SKIP      ^usr/lib/MACH(iconv)/geniconvtbl.so$ # 4384329

57 # picl file exclusions (4385799)
58 SKIP      ^usr/platform/.*/libpsvcplugin_psr\.so\.1
59 SKIP      ^usr/platform/.*/libpsvcpolicy_psr\.so\.1
60 SKIP      ^usr/platform/.*/libpsvcpolicy\.so\.1
```

## new/exception\_lists/check\_rtime

2

```
61 SKIP      ^usr/lib/sysevent/modules/picl_slm.so$

63 # Objects that are allowed to have executable data segments
64 EXEC_DATA    ^MACH(lib)/ld\.so\.1$          # 6524709, 32-bit, needed for x86 only
65 EXEC_DATA    ^lib/libc\.so\.1$               # 6524709, 32-bit, needed for x86 only
66 EXEC_DATA    ^opt/SUNWdrtt/tst/_*/ustack/tst\.helper\.exe$ # 6524709, 32-bit, needed for x86 only
67 EXEC_DATA    ^platform/*./MACH(kernel)/unix$ # 6524709, 32-bit, needed for x86 only
68 EXEC_DATA    ^platform/*./multiboot$ # 6524709, 32-bit, needed for x86 only

70 # Objects that are allowed to have an executable stack
71 EXEC_STACK   ^platform/*./MACH(kernel)/unix$ # 6524709, 32-bit, needed for x86 only
72 EXEC_STACK   ^platform/*./multiboot$ # 6524709, 32-bit, needed for x86 only

74 # Objects for which we allow relocations to the text segment
75 TEXTREL     ^platform/*./MACH(kernel)/unix$ # 6524709, 32-bit, needed for x86 only

77 # Directories and files that are allowed to have no direct bound symbols
78 NODIRECT    ^platform/*./MACH(kernel)/unix$ # 6524709, 32-bit, needed for x86 only
79 NODIRECT    ^usr/ucb # 6524709, 32-bit, needed for x86 only
80 NODIRECT    ^usr/4lib/sbcp$ # 6524709, 32-bit, needed for x86 only

82 # Identify any files that should be skipped when building a crle(1)
83 # configuration file. As the hwcap libraries can be loop-back mounted onto
84 # libc, these can confuse crle(1) because of their identical dev/inode.
85 NOCREALT   ^usr/lib/libc_hwcap[1-3].so.1$ # 6524709, 32-bit, needed for x86 only

87 # Files that should contain debugging information.
88 STAB        ^platform/*./MACH(kernel)/unix$ # 6524709, 32-bit, needed for x86 only

90 # Files that are allowed undefined references
91 UNDEF_REF   ^usr/lib/libnisdb\.so\.2$ # 6524709, 32-bit, needed for x86 only
92 UNDEF_REF   ^usr/snadm/lib/libsvm\.so\.1$ # 6524709, 32-bit, needed for x86 only

94 # Objects allowed to have unused dependencies
95 UNUSED_DEPS ^usr/lib/picl/plugins/          # require devtree dependencies
96 UNUSED_DEPS ^usr/lib/libp # profile libc makes libm an unused dep of libc

98 # libm.so.2 dependency
99 UNUSED_OBJ  unused object=.*MACH(libm)/libm_hwcap1\.so\.2 # 6524709, 32-bit, needed for x86 only

101 # Unused runpaths due to dlopen() use
102 UNUSED_RPATH /usr/lib/fs/autofs.*\ from\ .automountd # 6524709, 32-bit, needed for x86 only
103 UNUSED_RPATH /etc/ppp/plugins.*\ from\ .pppd # 6524709, 32-bit, needed for x86 only
104 UNUSED_RPATH /usr/lib/inet/ppp.*\ from\ .*pppd # 6524709, 32-bit, needed for x86 only
105 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libipsecutil\.so\.1 # 6524709, 32-bit, needed for x86 only
106 UNUSED_RPATH /usr/platform/*./rsmlib.*\ from\ .*librsm\.so\.2 # 6524709, 32-bit, needed for x86 only
107 UNUSED_RPATH \${ORIGIN.*}\ from\ .*fcode.so # 6524709, 32-bit, needed for x86 only
108 UNUSED_RPATH /opt/VRTSvxvm/lib.*\ from\ .*libdiskmgmt\.so\.1 # 6524709, 32-bit, needed for x86 only

110 # Unused runpaths in picl code
111 UNUSED_RPATH /usr/platform/*.*\ from\ .*usr/platform # 6524709, 32-bit, needed for x86 only
112 UNUSED_RPATH /usr/lib/picl/*.*\ from\ .*usr/platform # 6524709, 32-bit, needed for x86 only
113 UNUSED_RPATH /usr/platform/*.*\ from\ .*usr/lib/picl # 6524709, 32-bit, needed for x86 only

115 # Unused runpaths in non-OSNET objects we can't change
116 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libnss3\.so # 6524709, 32-bit, needed for x86 only
117 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libnssutil3\.so # 6524709, 32-bit, needed for x86 only
118 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libsmime3\.so # 6524709, 32-bit, needed for x86 only
119 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libssl3\.so # 6524709, 32-bit, needed for x86 only
120 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libdbus-1\.so\.3 # 6524709, 32-bit, needed for x86 only
121 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libdbus-glib-1\.so\.2 # 6524709, 32-bit, needed for x86 only
122 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libglib-2\.0\.so\.0 # 6524709, 32-bit, needed for x86 only
123 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libglib-2\.0\.so\.0 # 6524709, 32-bit, needed for x86 only
124 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libgobject-2\.0\.so\.0 # 6524709, 32-bit, needed for x86 only
125 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libgobject-2\.0\.so\.0 # 6524709, 32-bit, needed for x86 only
126 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libgthread-2\.0\.so\.0 # 6524709, 32-bit, needed for x86 only
```

```

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

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

143 # Unreferenced objects of non-OSnet objects we can't change
144 UNREF_OBJ       /lib.*\ of\ .*libcimapi\.so
145 UNREF_OBJ       /lib.*\ of\ .*libdbus-1\.so\.3
146 UNREF_OBJ       /lib.*\ of\ .*libdbus-glib-1\.so\.2
147 UNREF_OBJ       /lib.*\ of\ .*libglib-2.0\.so\.0
148 UNREF_OBJ       /lib.*\ of\ .*libgobject-2.0\.so\.0
149 UNREF_OBJ       /lib.*\ of\ .*libgthread-2\.0\.so\.0
150 UNREF_OBJ       /lib.*\ of\ .*libjvm\.so
151 UNREF_OBJ       /lib.*\ of\ .*libnetsnmp\.so\.15
152 UNREF_OBJ       /lib.*\ of\ .*libnetsnmpagent\.so\.15
153 UNREF_OBJ       /lib.*\ of\ .*libnetsnmpmibs\.so\.15
154 UNREF_OBJ       /lib.*\ of\ .*libnetsnmphelpers\.so\.15
155 UNREF_OBJ       /lib.*\ of\ .*libnspr4\.so
156 UNREF_OBJ       /lib.*\ of\ .*libpq\.so\.5
157 UNREF_OBJ       /lib.*\ of\ .*libsoftokn3\.so
158 UNREF_OBJ       /lib.*\ of\ .*libspmicommon\.so\.1
159 UNREF_OBJ       /lib.*\ of\ .*libspmocommon\.so\.1
160 UNREF_OBJ       /lib.*\ of\ .*libssl3\.so
161 UNREF_OBJ       /lib.*\ of\ .*libtspi\.so\.1
162 UNREF_OBJ       /lib.*\ of\ .*libxml2\.so\.2
163 UNREF_OBJ       /lib.*\ of\ .*libxslt\.so\.1
164 UNREF_OBJ       /lib.*\ of\ .*libpq\.so\.4
165 UNREF_OBJ       /lib.*\ of\ .*libpython2\.4\.so\.1\.0
166 UNREF_OBJ       /libgcc_s.*\ of\ .*libstdc\+\+\.so\.6

168 # Unreferenced object of objects we can't change for other reasons
169 UNREF_OBJ       /libmapmalloc\.so\.1; \ unused\ dependency\ of   # interposer
170 UNREF_OBJ       /libstdc\+\+\.so\.6; \ unused\ dependency\ of   # gcc build
171 UNREF_OBJ       /libm\.so\.2.*\ of\ .*libstdc\+\+\.so\.6      # gcc build
172 UNREF_OBJ       /lib.*\ of\ .*lib/picl/plugins/          # picl
173 UNREF_OBJ       /lib.*\ of\ .*kcfcd                  # interposer
174 UNREF_OBJ       /libpkcs11\.so\.1; \.*\ of\ .*libkmf\.so\.1  # interposed
175 # Referenced by the Studio build, not the GCC build.  GCC eliminates the unused
176 # statics which have the dependence.
177 UNREF_OBJ       /libc\.so\.1.*\ of\ .*ldap\.so\.1

180 # Objects that used to contain system functionality that has since
181 # migrated to libc. We preserve these libraries as pure filters for
182 # backward compatibility but nothing needs to link to them.
183 OLDDEP          libaio\.so\.1           # onnv build 44
184 OLDDEP          libdl\.so\.1            # on10 build 49
185 OLDDEP          libdoor\.so\.1          # onnv build 12
186 OLDDEP          libintl\.so\.1          # on297 build 7
187 OLDDEP          libpthread\.so\.1        # on10 build 53
188 OLDDEP          librt\.so\.1             # onnv build 44
189 OLDDEP          libsched\.so\.1          # on10 build 36
190 OLDDEP          libthread\.so\.1          # on10 build 53
191 OLDDEP          libw\.so\.1              # on297 build 7

```

```

193 # Files for which we skip checking of duplicate addresses in the
194 # symbol sort sections. Such exceptions should be rare --- most code will
195 # not have duplicate addresses, since it takes assembler or a "#pragma weak"
196 # to do such aliasing in C. C++ is different: The compiler generates aliases
197 # for implementation reasons, and the mangled names used to encode argument
198 # and return value types are difficult to handle well in mapfiles.
199 # Furthermore, the Sun compiler and gcc use different and incompatible
200 # name mangling conventions. Since ON must be buildable by either, we
201 # would have to maintain two sets of mapfiles for each such object.
202 # C++ use is rare in ON, so this is not worth pursuing.
203 #

204 NOSYMSORT      opt/SUNWdtrt/tst/common/pid/tst.weak2.exe      # DTrace test
205 NOSYMSORT      lib/amd64/libns1\.so\.1                      # C++
206 NOSYMSORT      lib/sparcv9/libns1\.so\.1                    # C++
207 NOSYMSORT      lib/sparcv9/libfru\.so\.1                   # C++
208 NOSYMSORT      usr/lib/sgml/nsgmls                         # C++
209 NOSYMSORT      usr/lib/lms                                # C++
210 NOSYMSORT      ld\.so\.1                                 # libc_pic.a use
211 NOSYMSORT      lib/libsun_fc\.so\.1                      # C++
212 NOSYMSORT      lib/amd64/libsun_fc\.so\.1                  # C++
213 NOSYMSORT      lib/sparcv9/libsun_fc\.so\.1                # C++
214 NOSYMSORT      usr/lib/amd64/libfru\.so\.1                # C++

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

```

### new/exception\_lists/interface\_check

```
*****
3705 Wed Aug 8 20:59:27 2012
new/exception_lists/interface_check
Bump Apache dependency to Apache 2
*****  
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/efcde/*\.so$  
49 PLUGIN      ^usr/lib/elfedit
50 PLUGIN      ^usr/lib/fm/fnd/plugins
51 PLUGIN      ^usr/lib/fm/fnd/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/pcl
58 PLUGIN      ^usr/lib/picl/plugins
59 PLUGIN      ^usr/lib/python2.[46]
60 PLUGIN      ^usr/lib/rcm/modules
```

### 1

### new/exception\_lists/interface\_check

```
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  
  
68 # sbcp is a special case, and not a plugin. However, it does not have a
69 # versioned name, and does not contain versioning, so the PLUGIN exemptions fit.
70 PLUGIN      ^usr/4lib/sbcp$  
  
73 # Objects that are not expected to contain versioning information.
74 # Note that PLUGIN objects are automatically exempt from this,
75 # so these directives are generally applied to non-plugin objects
76 NOVERDEF    ^usr/4lib/libc\.so\.
77 NOVERDEF    ^usr/MACH(lib)/0@\.so\.1$  
78 NOVERDEF    ^usr/lib/MACH(abi)/apptrace\.so\.1$  
79 NOVERDEF    ^usr/MACH(lib)/libfru.*\.so\.1$  
80 NOVERDEF    ^usr/MACH(lib)/libkrb5\.so\.1$  
81 NOVERDEF    ^usr/MACH(lib)/libzpool\.so\.1$  
82 NOVERDEF    ^usr/MACH(lib)/madv\.so\.1$  
83 NOVERDEF    ^usr/MACH(lib)/mpss\.so\.1$  
84 NOVERDEF    ^usr/MACH(lib)/s10_brand\.so\.1$  
85 NOVERDEF    ^usr/MACH(lib)/s10_npreload\.so\.1$  
86 NOVERDEF    ^usr/MACH(lib)/sn1_brand\.so\.1$  
87 NOVERDEF    ^usr/lib/fs/[/*]/fstyp\.so\.1$  
88 NOVERDEF    ^usr/lib/libmilter\.so\.1$  
89 NOVERDEF    ^usr/lib/libwrap\.so\.1\.0$  
90 NOVERDEF    ^usr/lib/locale/MACH(iso_8859_1)/iso_8859_1\.so\.3$  
91 NOVERDEF    ^usr/lib/picl/plugins$  
92 NOVERDEF    ^usr/sadm/admin/dhcpmgr/dhcpmgr\.so\.1$  
93 NOVERDEF    ^usr/sadm/admin/printmgr/lib/libpmgr\.so\.1$  
  
96 # Objects that are allowed to deviate from our standard version
97 # names.
98 NONSTD_VERNAM ^usr/MACH(lib)/libtecla\.so\.1$  
  
101 # These libc variants have an SONAME of libc\.so\.1$  
102 NONSTD_VERNAM ^usr/MACH(lib)/libc/libc_hwcap[1-3]+\..o\.1$  
  
105 # The ABI requires the SONAME for libsys.so.1 to be /usr/lib/ld.so.1
106 # That means that the base version will also be /usr/lib/ld.so.1, which
107 # is non-standard.
108 NONSTD_VERNAM ^usr/lib/libsys\.so\.1$
```

### 2

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

```
*****  
10285 Wed Aug 8 20:59:27 2012  
new/usr/src/cmd/logadm/glob.c  
Bump Apache dependency to Apache 2  
*****  
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 * this is basically the same syntax that csh supports for globs and  
36 * is provided by the routine glob_glob() which takes a filename and  
37 * returns a list of filenames that match the glob.  
38 *  
39 * the second type is something called a "reglob" which is a pathname  
40 * where the components are regular expressions as described in regex(3c).  
41 * some examples:  
42 *  
43 *      .*\.\.c  
44 *      /var/log/syslog\..  
45 *      log[0-9].*file  
46 *      /var/log/syslog\.( [0-9]+ )$0  
47 *  
48 * the last example uses the ()$n form to assign a numeric extension  
49 * on a filename to the "n" value kept by the fn routines with each  
50 * filename (see fn_setn() in fn.c). logadm uses this mechanism to  
51 * correctly sort lognames when templates containing $n are used.  
52 *  
53 * the routine glob_reglob() is used to expand reglobs. glob_glob()  
54 * is implemented by expanding the curly braces, converting the globs  
55 * to reglobs, and then passing the work to glob_reglob().  
56 *  
57 * finally, since expanding globs and reglobs requires doing a stat(2)  
58 * on the files, we store the resulting stat information in the filename  
59 * struct (see fn_setstat() in fn.c).  
60 *
```

1

new/usr/src/cmd/logadm/glob.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 #pragma ident "%Z%M% %I% %E% SMI"  
65 #include <stdio.h>  
66 #include <libintl.h>  
67 #include <stdlib.h>  
68 #include <libgen.h>  
69 #include <strings.h>  
70 #include <sys/types.h>  
71 #include <sys/param.h>  
72 #include <sys/stat.h>  
73 #include <dirent.h>  
74 #include "err.h"  
75 #include "fn.h"  
76 #include "glob.h"  
77 /* forward declarations for functions used internally by this module */  
78 static struct fn_list *glob_debrace(struct fn *fnp);  
79 static struct fn_list *glob_reglob_list(struct fn_list *fnlp);  
80 static boolean_t glob_magic(struct fn *fnp);  
81 static /* expand curly braces (like file{one,two,three}name) */  
82 static struct fn_list *  
83 glob_debrace(struct fn *fnp)  
84 {  
85     struct fn_list *ret = fn_list_new(NULL);  
86     struct fn_list *newret;  
87     char *sp = fn_s(fnp);  
88     char *left;  
89     char *right;  
90     char *comma;  
91  
92     /* start with an empty string in the list */  
93     fn_list_adds(ret, "");  
94  
95     /* while braces remain... */  
96     while (sp != NULL && (left = strchr(sp, '{')) != NULL)  
97         if ((right = strchr(left, '}')) == NULL){  
98             err(EF_FILE|EF_JMP, "Missing }");  
99             fn_list_free(ret);  
100            return (NULL);  
101        } else {  
102            /* stuff before "left" is finished */  
103            fn_list_appendrange(ret, sp, left);  
104  
105            /* stuff after "right" still need processing */  
106            sp = right + 1;  
107  
108            if (left + 1 == right)  
109                continue; /* just an empty {} */  
110  
111            /* stuff between "left" and "right" is comma-sep list */  
112            left++;  
113            newret = fn_list_new(NULL);  
114            while ((comma = strchr(left, ',')) != NULL) {  
115                struct fn_list *dup = fn_list_dup(ret);  
116  
117                /* stuff from left to comma is one variant */  
118                fn_list_appendrange(dup, left, comma);  
119                fn_list_addfn_list(newret, dup);  
120                left = comma + 1;  
121            }  
122  
123            /* what's left is the last item in the list */  
124        }
```

2

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

unchanged portion omitted

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

\*\*\*\*\*

34483 Wed Aug 8 20:59:28 2012

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

Bump Apache dependency to Apache 2

\*\*\*\*\*

\_\_\_\_\_ unchanged\_portion\_omitted \_\_\_\_\_

```
565 /* absurdly long buffer lengths for holding user/group/mode strings */
566 #define TIMESTRMAX    100
567 #define MAXATTR     100
568
569 /* rotate a log file if necessary, returns true if ok to go on to expire step */
570 static boolean_t
571 rotatelog(struct fn *fnp, struct opts *opts)
572 {
573     char *fname = fn_s(fnp);
574     struct stat stbuf;
575     char nowstr[TIMESTRMAX];
576     struct fn *recentlog = fn_new(NULL); /* for -R cmd */
577     char ownerbuf[MAXATTR];
578     char groupbuf[MAXATTR];
579     char modebuf[MAXATTR];
580     const char *owner;
581     const char *group;
582     const char *mode;
583
584     if (Debug && fname != NULL)
585         (void) fprintf(stderr, "rotatelog: fname <%s>\n", fname);
586
587     if (opts_count(opts, "p") && opts_optarg_int(opts, "p") == OOPTP_NEVER)
588         return (B_TRUE); /* "-p never" forced no rotate */
589
590     /* prepare the keywords */
591     kw_init(fnp, NULL);
592     if (Debug > 1) {
593         (void) fprintf(stderr, "rotatelog keywords:\n");
594         kw_print(stderr);
595     }
596
597     if (lstat(fname, &stbuf) < 0) {
598         if (opts_count(opts, "N"))
599             return (1);
600         err(EF_WARN|EF_SYS, "%s", fname);
601     }
602     return (B_FALSE);
603
604     if ((stbuf.st_mode & S_IFMT) == S_IFLNK) {
605         err(EF_WARN, "%s is a symlink", fname);
606     }
607     return (B_FALSE);
608
609     if ((stbuf.st_mode & S_IFMT) != S_IFREG) {
610         err(EF_WARN, "%s is not a regular file", fname);
611     }
612     return (B_FALSE);
613
614     /* even if size condition is not met, this entry is "done" */
615     if (opts_count(opts, "s") &&
616         stbuf.st_size < opts_optarg_int(opts, "s")) {
617         Donenames = lut_add(Donenames, fname, "1");
618         return (B_TRUE);
619     }
620
621     /* see if age condition is present, and return if not met */
622     if (opts_count(opts, "p")) {
```

1

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

```
623
624     off_t when = opts_optarg_int(opts, "p");
625     struct opts *cfopts;
626
627     /* unless rotate forced by "-p now", see if period has passed */
628     if (when != OOPTP_NOW) {
629         /*
630          * "when" holds the number of seconds that must have
631          * passed since the last time this log was rotated.
632          * of course, running logadm can take a little time
633          * (typically a second or two, but longer if the
634          * conffile has lots of stuff in it) and that amount
635          * of time is variable, depending on system load, etc.
636          * so we want to allow a little "slop" in the value of
637          * "when". this way, if a log should be rotated every
638          * week, and the number of seconds passed is really a
639          * few seconds short of a week, we'll go ahead and
640          * rotate the log as expected.
641         */
642         if (when >= 60 * 60)
643             when -= 59;
644
645         /*
646          * last rotation is recorded as argument to -P,
647          * but if logname isn't the same as log file name
648          * then the timestamp would be recorded on a
649          * separate line in the conf file. so if we
650          * haven't seen a -P already, we check to see if
651          * it is part of a specific entry for the log
652          * file name. this handles the case where the
653          * logname is "apache", it supplies a log file
654          * name like "/var/apache2/2.2/logs/[a-z]*_log",
655          * which expands to multiple file names. if one of
656          * the file names is "/var/apache2/2.2/logs/access_log"
657          * then the -P will be attached to a line with that
658          * name like "/var/apache/logs/[a-z]*_log",
659          * which expands to multiple file names. if one
660          * of the file names is "/var/apache/logs/access_log"
661          * the the -P will be attached to a line with that
662          * logname in the conf file.
663         */
664         if (opts_count(opts, "P")) {
665             off_t last = opts_optarg_int(opts, "P");
666
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             /*
675              * just checking this means this entry
676              * is now "done" if we're going through
677              * the entire conffile
678             */
679             Donenames = lut_add(Donenames, fname, "1");
680
681             /*
682              * return if not enough time has passed
683              */
684             if (Debug)
```

2

```

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

```

```

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

---

*unchanged portion omitted*

```
*****
69918 Wed Aug 8 20:59:29 2012
new/usr/src/cmd/logadm/tester
Bump Apache dependency to Apache 2
*****
_____unchanged_portion_omitted_____
256 #####
257 ##### THE TESTS START AFTER HERE...
259 #####
260 #####
262 # common setup step -- create a testfile.conf
263 sub set_testconffile {
264     my $fname = shift;
265     $fname = 'testfile.conf' unless defined($fname);
266     set_file($fname, <<'EOF');
267 #
268 # logadm.conf
270 #
271 # Default settings for system log file management.
272 # The -w option to logadm(1M) is the preferred way to write to this file,
273 # but if you do edit it by hand, use "logadm -V" to check it for errors.
274 #
275 # The format of lines in this file is:
276 #       <logname> <options>
277 # For each logname listed here, the default options to logadm
278 # are given. Options given on the logadm command line override
279 # the defaults contained in this file.
280 #
281 # logadm typically runs early every morning via an entry in
282 # root's crontab (see crontab(1)).
283 #
284 /var/adm/messages -C 4 -P 'Thu Nov 1 16:56:42 2001' -a 'kill -HUP `cat /var/run
285 /var/cron/log -s 512k -t /var/cron/olog
286 /var/lp/logs/lpsched -C 2 -N -t '$file.$N'
287 #
288 # The entry below is used by turnacct(1M)
289 #
290 /var/adm/pacct -C 0 -a '/usr/lib/acct/accton pacct' -g adm -m 664 -o adm -p neve
291 apache -C 24 -a '/usr/apache2/2.2/bin/apachectl graceful' -p lm -t '/var/apache2
291 apache -C 24 -a '/usr/apache/bin/apachectl graceful' -p lm -t '/var/apache/old-1
292 /var/log/syslog -C 8 -P 'Thu Nov 1 09:16:38 2001' -a 'kill -HUP `cat /var/run/s
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 /var/apache/logs/access_log -P 'Thu Nov 1 08:27:56 2001'
299 /var/apache/logs/error_log -P 'Thu Nov 1 08:27:56 2001'
299 /var/apache/logs/suexec_log -P 'Thu Nov 1 08:27:56 2001'
299 /var/apache/logs/mod_jserv.log -P 'Thu Nov 1 08:27:56 2001'
299 EOF
299 }
_____unchanged_portion_omitted_____
812 #####
813 #
814 #      logadmV1 -- test of "logadm -V"
815 #
816 #####
817 sub logadmV1 {
818     set_testconffile;
```

```
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
825 apache -C 24 -a '/usr/apache/bin/apachectl graceful' -p lm -t '/var/apache/old-1
826 /var/log/syslog -C 8 -P 'Thu Nov 1 09:16:38 2001' -a 'kill -HUP `cat /var/run/s
827 /var/apache2/2.2/logs/access_log -P 'Thu Nov 1 08:27:56 2001'
828 /var/apache2/2.2/logs/error_log -P 'Thu Nov 1 08:27:56 2001'
829 /var/apache2/2.2/logs/suexec_log -P 'Thu Nov 1 08:27:56 2001'
830 /var/apache2/2.2/logs/mod_jserv.log -P 'Thu Nov 1 08:27:56 2001'
831 /var/apache2/2.2/logs/jserv.log -P 'Thu Nov 1 08:27:56 2001'
832 /var/apache/logs/access_log -P 'Thu Nov 1 08:27:56 2001'
827 /var/apache/logs/error_log -P 'Thu Nov 1 08:27:56 2001'
828 /var/apache/logs/suexec_log -P 'Thu Nov 1 08:27:56 2001'
830 /var/apache/logs/mod_jserv.log -P 'Thu Nov 1 08:27:56 2001'
831 /var/apache/logs/jserv.log -P 'Thu Nov 1 08:27:56 2001'
832 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_____
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 }
```

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

```
*****
4199 Wed Aug 8 20:59:30 2012
new/usr/src/cmd/lp/cmd/lpsched/print-svc
Bump Apache dependency to Apache 2
*****
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
32SERVICE='echo $SMF_FMRI | /usr/bin/cut -f1,2 -d":"
34 case "$1" in
35 'start')
37 isoopts='/usr/sbin/svccfg <-EOF
38 select $SMF_FMRI
39 listpg cmd_opts
41 EOF'
43 if [ "$isoopts" ] ; 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
```

```
1
```

```
new/usr/src/cmd/lp/cmd/lpsched/print-svc
*****
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"
123     IPP_CONF="/etc/apache/httpd-standalone-ipp.conf"
124     ip="any"
126     policy='get_policy $FMRI'
```

```
2
```

```
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='svcprom -p inetd/name ${RFC1179_FMRI} 2>/dev/null'
133         rfc_proto='svcprom -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
138
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' ${IPP_CONF} | awk '{print
149
150             if [ -z "$ipp_ports" ]; then
151                 ipp_ports='grep ^[ \t]*[^# ]*Port' ${IPP_CONF} | \
152                     awk '{print $2}' | tail -1'
153         fi
154
155         for port in $ipp_ports; do
156             generate_rules $FMRI $policy "tcp" $ip $port $file
157         done
158     fi
159
160     ;;
161
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
```

```
*****
```

```
14678 Wed Aug 8 20:59:31 2012
```

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

```
Bump Apache dependency to Apache 2
```

```
*****
```

```
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 2012 Nexenta Systems, Inc. All rights reserved.  
24  * Copyright 2008 Sun Microsystems, Inc. All rights reserved.  
25  * Use is subject to license terms.  
26  *  
27 */  
  
29 /* $Id: ipp-listener.c 146 2006-03-24 00:26:54Z njacobs $ */  
  
30 #pragma ident "%Z%%M% %I%     %E% SMI"
```

```
31 #include <stdio.h>  
32 #include <stdlib.h>  
33 #include <string.h>  
34 #include <netinet/in.h>  
35 #include <assert.h>  
36 #include <errno.h>  
37 #include <syslog.h>  
38 #include <sys/types.h>  
39 #include <sys/stat.h>  
40 #include <fcntl.h>  
41 #include <unistd.h>  
42 #include <sys/systeminfo.h>  
  
44 #include <papi.h>  
45 #include <ipp-listener.h>  
46 #include <uri.h>  
  
48 typedef papi_status_t (ipp_handler_t)(papi_service_t svc,  
49                                         papi_attribute_t **request,  
50                                         papi_attribute_t ***response,  
51                                         ipp_reader_t iread, void *fd);  
  
53 /*  
54  * protocol request handlers are inserted below. The handler must be  
55  * declared extern immediately below this comment and then an entry  
56  * must be inserted in the "handlers" table a little further down.  
57 */  
58 extern ipp_handler_t ipp_print_job;  
59 extern ipp_handler_t ipp_validate_job;
```

```
1
```

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

```
*****
```

```
60 extern ipp_handler_t ipp_create_job;  
61 extern ipp_handler_t ipp_get_printer_attributes;  
62 extern ipp_handler_t ipp_get_jobs;  
63 extern ipp_handler_t ipp_pause_printer;  
64 extern ipp_handler_t ipp_resume_printer;  
65 extern ipp_handler_t ipp_disable_printer;  
66 extern ipp_handler_t ipp_enable_printer;  
67 extern ipp_handler_t ipp_purge_jobs;  
68 extern ipp_handler_t ipp_send_document;  
69 extern ipp_handler_t ipp_cancel_job;  
70 extern ipp_handler_t ipp_get_job_attributes;  
71 extern ipp_handler_t ipp_release_job;  
72 extern ipp_handler_t ipp_hold_job;  
73 extern ipp_handler_t ipp_restart_job;  
74 extern ipp_handler_t ipp_set_job_attributes;  
75 extern ipp_handler_t ipp_set_printer_attributes;  
76 extern ipp_handler_t cups_get_default;  
77 extern ipp_handler_t cups_get_printers;  
78 extern ipp_handler_t cups_get_classes;  
79 extern ipp_handler_t cups_accept_jobs;  
80 extern ipp_handler_t cups_reject_jobs;  
81 extern ipp_handler_t cups_move_job;  
  
83 /* ARGUSED0 */  
84 static papi_status_t  
85 default_handler(papi_service_t svc, papi_attribute_t **request,  
86                  papi_attribute_t ***response, ipp_reader_t iread, void *fd)  
87 {  
88     int result = (int)PAPI_INTERNAL_ERROR;  
  
89     if (response != NULL)  
90         (void) papiAttributeListGetInteger(*response, NULL,  
91                                             "status-code", &result);  
  
92     return ((papi_status_t)result);  
93 }  
94 unchanged_portion_omitted  
  
152 static int  
153 ipp_operation_name_to_index(const char *name)  
154 ipp_operation_name_to_index(char *name)  
154 {  
155     int i;  
  
157     for (i = 0; handlers[i].name != NULL; i++)  
158         if (strcasecmp(name, handlers[i].name) == 0)  
159             return (i);  
  
161     return (-1);  
162 }  
163 unchanged_portion_omitted  
  
234 static char  
235 type_to_boolean(const char *type)  
236 type_to_boolean(char *type)  
236 {  
237     char result = PAPI_FALSE;  
  
239     if ((strcasecmp(type, "true") == 0) ||  
240         (strcasecmp(type, "yes") == 0) ||  
241         (strcasecmp(type, "on") == 0) ||  
242         (strcasecmp(type, "enable") == 0))  
243         result = PAPI_TRUE;  
  
245     return (result);  
246 }  
247 unchanged_portion_omitted
```

```
2
```

```
277 papi_status_t
278 ipp_configure_operation(papi_attribute_t ***list, const char *operation, const c
279 ipp_configure_operation(papi_attribute_t ***list, char *operation, char *type)
279 {
280     papi_status_t result = PAPI_OPERATION_NOT_SUPPORTED;
281     char boolean = PAPI_FALSE;
283     if ((list == NULL) || (operation == NULL) || (type == NULL))
284         return (PAPI_BAD_ARGUMENT);
286     boolean = type_to_boolean(type);
288     if (strcasecmp(operation, "all") == 0) {
289         result = ipp_configure_all_operations(list, boolean);
290     } else if (strcasecmp(operation, "required") == 0) {
291         result = ipp_configure_required_operations(list, boolean);
292     } else if (ipp_operation_name_to_index(operation) != -1) {
293         result = papiAttributeListAddBoolean(list, PAPI_ATTR_REPLACE,
294                                              operation, boolean);
295     }
297 }
298 }  
unchanged portion omitted
```

```

new/usr/src/lib/print/libipp-listener/common/ipp-listener.h      1
*****
2355 Wed Aug  8 20:59:32 2012
new/usr/src/lib/print/libipp-listener/common/ipp-listener.h
Bump Apache dependency to Apache 2
*****
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 2012 Nexenta Systems, Inc. All rights reserved.
23 * Copyright 2006 Sun Microsystems, Inc. All rights reserved.
24 * Use is subject to license terms.
25 *
26 */
27 */
28 #ifndef _IPP_LISTENER_H
29 #define _IPP_LISTENER_H
30
31 /* $Id: ipp-listener.h 146 2006-03-24 00:26:54Z njacobs $ */
32
33 #pragma ident "%Z%%M% %I%     %E% SMI"
34
35 #ifdef __cplusplus
36 extern "C" {
37#endif
38
39 #include <ipp.h>
40
41 /* exported functions */
42 extern papi_status_t ipp_configure_operation(papi_attribute_t ***list,
43                                              const char *operation,
44                                              const char *type);
45 extern papi_status_t ipp_process_request(papi_attribute_t **request,
46                                         papi_attribute_t ***response,
47                                         ipp_reader_t iread, void *fd);
48
49 /* shared internal functions */
50 extern char *ipp_svc_status_msg(papi_service_t svc, papi_status_t status);
51 extern char *destination_from_printer_uri(char *);
52 extern void get_printer_id(papi_attribute_t **attributes, char **printer,
53                            int *id);
54 extern void ipp_operations_supported(papi_attribute_t ***list,
55                                     papi_attribute_t **request);
56 extern void get_string_list(papi_attribute_t **attributes, char *name,
57                            char ***values);
58 extern void add_default_attributes(papi_attribute_t ***attributes);
59 extern void papi_to_ipp_printer_group(papi_attribute_t ***response,
60                                     papi_attribute_t **request, int flags,
61                                     papi_printer_t p);
62 extern void papi_to_ipp_job_group(papi_attribute_t ***response,
63                                     papi_attribute_t **request, int flags, papi_job_t j);
64 extern void massage_response(papi_attribute_t **request,
65                             papi_attribute_t **response);
66 #ifdef __cplusplus
67 }
68 */

```

```

new/usr/src/lib/print/libipp-listener/common/ipp-listener.h      2
*****
59                                     papi_attribute_t **request, int flags,
60                                     papi_printer_t p);
61 extern void papi_to_ipp_job_group(papi_attribute_t ***response,
62                                     papi_attribute_t **request, int flags, papi_job_t j);
63 extern void massage_response(papi_attribute_t **request,
64                             papi_attribute_t **response);
65
66 #ifdef __cplusplus
67 }
68 */

```

```
new/usr/src/lib/print/libpapi-common/common/attribute.c
```

```
*****
25631 Wed Aug 8 20:59:33 2012
new/usr/src/lib/print/libpapi-common/common/attribute.c
Bump Apache dependency to Apache 2
*****
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 2012 Nexenta Systems, Inc. All rights reserved.
23 * Copyright 2008 Sun Microsystems, Inc. All rights reserved.
24 * Use is subject to license terms.
25 *
26 */
27 */
28 /* $Id: attribute.c 157 2006-04-26 15:07:55Z ktou $ */
29
30 #pragma ident "%Z%%M% %I%     %E% SMI"
31 /*LINTLIBRARY*/
32
33 #include <stdio.h>
34 #include <stdlib.h>
35 #include <stdarg.h>
36 #include <string.h>
37 #include <ctype.h>
38 #include <alloca.h>
39 #include <papi.h>
40 #include <regex.h>
41
42 #define MAX_PAGES 32767
43 /*
44 * Assuming the maximum number of pages in
45 * a document to be 32767
46 */
47
48 static void papiAttributeFree(papi_attribute_t *attribute);
49
50 static void
51 papiAttributeValueFree(papi_attribute_value_type_t type,
52                       papi_attribute_value_t *value)
53 {
54     if (value != NULL) {
55         switch (type) {
56             case PAPI_STRING:
57                 if (value->string != NULL)
58                     free(value->string);
59         break;
59 }
```

```
1
```

```
new/usr/src/lib/print/libpapi-common/common/attribute.c
```

```
2
60         case PAPI_COLLECTION:
61             if (value->collection != NULL) {
62                 int i;
63
64                 for (i = 0; value->collection[i] != NULL; i++)
65                     papiAttributeFree(value->collection[i]);
66
67                 free(value->collection);
68             }
69             break;
70         default: /* don't need to free anything extra */
71             break;
72     }
73
74     free(value);
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 static papi_attribute_t *
205 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;
209
210     if ((result = calloc(1, sizeof (*result))) != NULL) {
211         result->name = strdup(name);
212         result->type = type;
213     }
214
215     return (result);
216 }
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238 papi_status_t
239 papiAttributeListAddValue(papi_attribute_t ***list, int flgs,
240                           const char *name, papi_attribute_value_type_t type,
241                           char *name, papi_attribute_value_type_t type,
242                           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;
248
249     if ((list == NULL) || (name == NULL))
250         return (PAPI_BAD_ARGUMENT);
251
252     if ((type == PAPI_RANGE) && (value != NULL) &&
253         (value->range.lower > value->range.upper))
254         return (PAPI_BAD_ARGUMENT); /* RANGE must have min <= max */
255
256     if (flags == 0) /* if it wasn't set, set a default behaviour */
257         flags = PAPI_ATTR_APPEND;
258
259     /* look for an existing one */
260     attribute = papiAttributeListFind(*list, name);
261
262     if (((flags & PAPI_ATTR_EXCL) != 0) && (attribute != NULL))
263         return (PAPI_CONFLICT); /* EXISTS */
264
265     if (((flags & PAPI_ATTR_REPLACE) == 0) && (attribute != NULL) &&
266         (attribute->type != type))
267         return (PAPI_CONFLICT); /* TYPE CONFLICT */
268
269     /* if we don't have one, create it and add it to the list */
270 }
```

```
2
```

```
269     if ((attribute == NULL) &&
270         ((attribute = papiAttributeAlloc(name, type)) != NULL))
271         list_append(list, attribute);
273     /* if we don't have one by now, it's most likely an alloc fail */
274     if (attribute == NULL)
275         return (PAPI_TEMPORARY_ERROR);
277     /*
278      * if we are replacing, clear any existing values, but don't free
279      * until after we have replaced the values, in case we are replacing
280      * a collection with a relocated version of the original collection.
281      */
282     if (((flags & PAPI_ATTR_REPLACE) != 0) && (attribute->values != NULL)) {
283         values = attribute->values;
284         attribute->values = NULL;
285     }
287     attribute->type = type;
289     result = papiAttributeListAppendValue(&attribute->values, type, value);
291     /* free old values if we replaced them */
292     if (values != NULL)
293         papiAttributeValuesFree(type, values);
295     return (result);
296 }
unchanged_portion_omitted_
318 papi_status_t
319 papiAttributeListAddBoolean(papi_attribute_t ***list, int flags,
320                             const char *name, char boolean)
321 {
322     papi_attribute_value_t v;
324     v.boolean = boolean;
325     return (papiAttributeListAddValue(list, flags, name, PAPI_BOOLEAN, &v));
326 }
unchanged_portion_omitted_
403 papi_attribute_t *
404 papiAttributeListFind(papi_attribute_t **list, const char *name)
405 papiAttributeListFind(papi_attribute_t **list, char *name)
405 {
406     int i;
407     if ((list == NULL) || (name == NULL))
408         return (NULL);
410     for (i = 0; list[i] != NULL; i++)
411         if (strcasecmp(list[i]->name, name) == 0)
412             return ((papi_attribute_t *)list[i]);
414     return (NULL);
415 }
unchanged_portion_omitted_
```

new/usr/src/lib/print/libpapi-common/common/papi.h

1

```

*****
15768 Wed Aug  8 20:59:34 2012
new/usr/src/lib/print/libpapi-common/common/papi.h
Bump Apache dependency to Apache 2
*****
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 2012 Nexenta Systems, Inc. All rights reserved.
23  * Copyright 2008 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25  *
26  *
27 */
28 #ifndef _PAPI_H
29 #define _PAPI_H
30
31 /* $Id: papi.h 161 2006-05-03 04:32:59Z njacobs $ */
32
33 #pragma ident "%Z%%M% %I%      %E% SMI"
34
35 #include <sys/types.h>
36 #include <time.h>
37 #include <stdio.h>
38
39 #ifdef __cplusplus
40 extern "C" {
41 #endif
42 /*
43  * Types
44 */
45
46 /*      service related types */
47 typedef void *papi_service_t;
48 typedef void *papi_printer_t;
49 typedef void *papi_job_t;
50 typedef void *papi_stream_t;
51
52 typedef enum {
53     PAPI_ENCRYPT_IF_REQUESTED,          /* Encrypt if requested (TLS upgrade) */
54     PAPI_ENCRYPT_NEVER,                /* Never encrypt */
55     PAPI_ENCRYPT_REQUIRED,             /* Encryption required (TLS upgrade) */
56     PAPI_ENCRYPT_ALWAYS               /* Always encrypt (SSL) */
57 } papi_encryption_t;
58
59 /* unchanged portion omitted */

```

new/usr/src/lib/print/libpapi-common/common/papi.h

```

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

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

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

```

```

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

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

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

```

```
419 extern papi_status_t papiJobStreamAdd(papi_service_t handle, char *printer,
420                                         int32_t id, papi_stream_t *stream);
421 extern papi_status_t papiJobCommit(papi_service_t handle, char *printer,
422                                     int32_t id);
423 extern papi_status_t papiServiceSetPeer(papi_service_t handle, int peerfd);
424 #endif /* SOLARIS_PRIVATE_POST_0_9 */
426 extern char *papiStatusString(papi_status_t status);
428 /*
429  * Internal functions that aren't in the API, but are shared across
430  * protocol support implementations(psms) and the tightly bound
431  * listener library. Do not use these in your applications.
432 */
433 extern void list_append();
434 extern void list_concatenate();
435 extern void list_remove();
436 extern void copy_attributes(papi_attribute_t ***result,
437                             papi_attribute_t **list);
438 extern void split_and_copy_attributes(char **list,
439                                       papi_attribute_t **attributes,
440                                       papi_attribute_t ***in,
441                                       papi_attribute_t ***out);
443 extern papi_attribute_t **getprinterbyname(char *name, char *ns);
445 extern int is_localhost(char *hostname);
447 #ifdef __cplusplus
448 }
```

*unchanged\_portion omitted*

```

new/usr/src/lib/print/mod_ipp/Makefile
*****
2673 Wed Aug 8 20:59:35 2012
new/usr/src/lib/print/mod_ipp/Makefile
Bump Apache dependency to Apache 2
*****
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 2012 Nexenta Systems, Inc. All rights reserved.
23 # Copyright 2010 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #

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

31 include ../../Makefile.lib
32 include ../../Makefile.rootfs

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

38 ROOTDIRS = $(ROOT)/usr/apache2/2.2 $(APACHEMODDIR) $(APACHECONFDIR) \
37 ROOTDIRS = $(ROOT)/usr/apache $(APACHEMODDIR) $(APACHECONFDIR) \
39 $(ROOT)/var/lp $(LISTENERDIR)

41 $(ROOT)/var/lp:= DIRMODE = 775
42 $(ROOT)/var/lp:= FILEMODE = 775

44 LIBS =
45 $(DYNLIB)

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

48 APR_INC.cmd = apr-1-config --includes
49 APU_INC.cmd = apu-1-config --includes
50 CFLAGS += $(CCVERBOSE) -erroff=_IMPLICIT_DECL_FUNC_RETURN_INT

48 CFLAGS += $(CCVERBOSE)
51 CPPFLAGS += -I../../libipp-listener/common
52 CPPFLAGS += -I../../libipp-core/common
53 CPPFLAGS += -I/usr/apache2/2.2/include
54 CPPFLAGS += $(APR_INC.cmd:sh)
55 CPPFLAGS += $(APU_INC.cmd:sh)
51 CPPFLAGS += -I/usr/apache/include

```

```

1
2 new/usr/src/lib/print/mod_ipp/Makefile
56 CPPFLAGS += -DEAPI
57 ZDEFS = $(ZNODEFS)
59 MAPFILES = mapfile
61 LDLIBS += -lipp-listener -lipp-core -lpapi -lc
63 # SMF manifest
64 MANIFEST= ipp-listener.xml
65 ROOTMANIFESTDIR= $(ROOT)/lib/svc/manifest/application/print
66 ROOTMANIFEST= $(MANIFEST:%= $(ROOTMANIFESTDIR)/%)
67 $(ROOTMANIFEST) := FILEMODE = 444
69 # Apache module
70 $(APACHEMODDIR)/$(LIBLINKS): $(ROOTDIRS)
72 # Apache config
73 APACHECONFFILE= $(APACHECONFDIR)/httpd-standalone-ipp.conf
74 $(APACHECONFFILE) := FILEMODE = 644
75 LISTENERFILE= $(LISTENERDIR)/index.html
76 $(LISTENERFILE) := FILEMODE = 444
78 $(ROOT)/var/lp:= FILEMODE = 0775
80 $(APACHEMODDIR)/$(LIBLINKS):= FILEMODE = 0555
82 $(ROOTMANIFESTDIR)/% $(APACHEMODDIR)/% $(APACHECONFDIR)/% $(LISTENERDIR)/%:
83 $(INS.file)
85 $(ROOTDIRS):
86 $(INS.dir)
88 .KEEP_STATE:
90 all: $(LIBS)
92 install: all $(APACHEMODDIR)/$(LIBLINKS) $(APACHECONFFILE) \
93 $(LISTENERFILE) $(ROOTMANIFEST)
95 install_h:
97 lint:
99 include ../../Makefile.targ

```

```
new/usr/src/lib/print/mod_ipp/mod_ipp.c
```

```
*****
```

```
14473 Wed Aug 8 20:59:36 2012
```

```
new/usr/src/lib/print/mod_ipp/mod_ipp.c
```

```
Bump Apache dependency to Apache 2
```

```
*****
```

```
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 2012 Nexenta Systems, Inc. All rights reserved.
24  * Copyright 2006 Sun Microsystems, Inc. All rights reserved.
25  * Use is subject to license terms.
26  */
27 */
```

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

```
30 #pragma ident "%Z%%M% %I%     %E% SMI"
```

```
31 /*
32  * Internet Printing Protocol (IPP) module for Apache.
33 */
```

```
35 #include "ap_config.h"
```

```
37 #include <stdio.h>
38 #include <time.h>
39 #include <sys/time.h>
40 #include <values.h>
41 #include <libintl.h>
42 #include <alloca.h>
```

```
44 #include "httpd.h"
45 #include "http_config.h"
46 #include "http_core.h"
47 #include "http_protocol.h"
48 #include "http_log.h"
49 #include "http_main.h"
50 #include "papi.h"
51 #ifndef APACHE_RELEASE /* appears to only exist in Apache 1.X */
52 #define APACHE2
53 #define ap_table_get apr_table_get
54 #include "apr_compat.h"
55 #endif
```

```
56 #include <papi.h>
57 #include <ipp-listener.h>
```

```
1
```

```
new/usr/src/lib/print/mod_ipp/mod_ipp.c
```

```
59 #ifndef APACHE2
60 module MODULE_VAR_EXPORT ipp_module;
61 #else
62 module AP_MODULE_DECLARE_DATA ipp_module;
63 #endif

65 #ifndef AP_INIT_TAKE1 /* Apache 2.X has this, but 1.3.X does not */
66 #define AP_INIT_NO_ARGS(directive, action, arg, where, mesg) \
67 { directive, action, arg, where, NO_ARGS, mesg }
68 #define AP_INIT_TAKE1(directive, action, arg, where, mesg) \
69 { directive, action, arg, where, TAKE1, mesg }
70 #define AP_INIT_TAKE2(directive, action, arg, where, mesg) \
71 { directive, action, arg, where, TAKE2, mesg }
72#endif

74 typedef struct {
75     int conformance;
76     char *default_user;
77     char *default_svc;
78     papi_attribute_t **operations;
79 } IPPListenerConfig;
unchanged_portion_omitted

178 void _log_error(const char *file, int line, int level, request_rec *r,
179                  const char *fmt, ...)
180 {
181     va_list args;
182     size_t size;
183     char *message = alloca(BUFSIZ);
184
185     va_start(args, fmt);
186     /*
187      * fill in the message. If the buffer is too small, allocate
188      * one that is large enough and fill it in.
189      */
190     if ((size = vsnprintf(message, BUFSIZ, fmt, args)) >= BUFSIZ)
191         if ((message = alloca(size)) != NULL)
192             vsnprintf(message, size, fmt, args);
193     va_end(args);

195 #ifdef APACHE2
196     ap_log_error(file, line, level, APR_SUCCESS, r, message);
197 #else
198     ap_log_error(file, line, level, NULL, r, message);
199 #endif
200 }

202 static int
203 ipp_handler(request_rec *r)
204 {
205     papi_attribute_t **request = NULL, **response = NULL;
206     IPPListenerConfig *config;
207     papi_status_t status;
208     int ret;
209
210     /* Really, IPP is all POST requests */
211     if (r->method_number != M_POST)
212         return (DECLINED);

214 #ifndef APACHE2
215     /*
216      * An IPP request must have a MIME type of "application/ipp"
217      * (RFC-2910, Section 4, page 19). If it doesn't match this
218      * MIME type, we should decline the request and let someone else
219      * try and handle it.
220 
```

```
2
```

```

220         */
221     if (r->headers_in != NULL) {
222         char *mime_type = (char *)ap_table_get(r->headers_in,
223                                             "Content-Type");
224
225         if ((mime_type == NULL) ||
226             (strcasecmp(mime_type, "application/ipp") != 0))
227             return (DECLINED);
228     }
229 #endif
230 /* CHUNKED_DECHUNK might not work right for IPP? */
231 if ((ret = ap_setup_client_block(r, REQUEST_CHUNKED_DECHUNK)) != OK)
232     return (ret);
233
234     if (!ap_should_client_block(r))
235         return (HTTP_INTERNAL_SERVER_ERROR);
236
237 #ifndef APACHE2
238     ap_soft_timeout("ipp_module: read/reply request ", r);
239 #endif
240 /* read the IPP request off the network */
241 status = ipp_read_message(read_data, r, &request, IPP_TYPE_REQUEST);
242
243     if (status != PAPI_OK)
244         _log_error(APLOG_MARK, APLOG_ERR, r,
245                    "read failed: %s\n", papiStatusString(status));
246 #ifdef DEBUG
247     papiAttributeListPrint(stderr, request, "request (%d) ", getpid());
248 #endif
249
250     (void) papiAttributeListAddString(&request, PAPI_ATTR_EXCL,
251                                     "originating-host", (char *));
252 #ifdef APACHE2
253     ap_get_remote_host
254         (r->connection, r->per_dir_config, REMOTE_NAME, NULL));
255 #else
256     ap_get_remote_host
257         (r->connection, r->per_dir_config, REMOTE_NAME));
258 #endif
259
260     (void) papiAttributeListAddInteger(&request, PAPI_ATTR_EXCL,
261                                         "uri-port", ap_get_server_port(r));
262     if (r->headers_in != NULL) {
263         char *host = (char *)ap_table_get(r->headers_in, "Host");
264
265         if ((host == NULL) || (host[0] == '\0'))
266             host = (char *)ap_get_server_name(r);
267
268         (void) papiAttributeListAddString(&request, PAPI_ATTR_EXCL,
269                                         "uri-host", host);
270     }
271     (void) papiAttributeListAddString(&request, PAPI_ATTR_EXCL,
272                                         "uri-path", r->uri);
273
274     config = ap_get_module_config(r->per_dir_config, &ipp_module);
275     if (config != NULL) {
276         (void) papiAttributeListAddInteger(&request, PAPI_ATTR_EXCL,
277                                         "conformance", config->conformance);
278         (void) papiAttributeListAddCollection(&request, PAPI_ATTR_EXCL,
279                                         "operations", config->operations);
280         if (config->default_user != NULL)
281             (void) papiAttributeListAddString(&request,
282                                         PAPI_ATTR_EXCL, "default-user",
283                                         config->default_user);
284         if (config->default_svc != NULL)
285             (void) papiAttributeListAddString(&request,

```

```

286
287
288     }
289
290 #ifndef APACHE2
291     /*
292      * For Trusted Solaris, pass the fd number of the socket connection
293      * to the backend so the it can be forwarded to the backend print
294      * service to retrieve the sensitivity label off of a multi-level
295      * port.
296      */
297     (void) papiAttributeListAddInteger(&request, PAPI_ATTR_EXCL,
298                                         "peer-socket", ap_bfileno(r->connection->client, B_RD));
299 #endif
300
301     /* process the request */
302     status = ipp_process_request(request, &response, read_data, r);
303     if (status != PAPI_OK) {
304         errno = 0;
305         _log_error(APLOG_MARK, APLOG_ERR, r,
306                    "request failed: %s\n", papiStatusString(status));
307         discard_data(r);
308     }
309 #ifdef DEBUG
310     fprintf(stderr, "processing result: %s\n", papiStatusString(status));
311     papiAttributeListPrint(stderr, response, "response (%d) ", getpid());
312 #endif
313
314     /*
315      * If the client is using chunking and we have not yet received the
316      * final "0" sized chunk, we need to discard any data that may
317      * remain in the post request.
318      */
319     if ((r->read_chunked != 0) &&
320         (ap_table_get(r->headers_in, "Content-Length") == NULL))
321         discard_data(r);
322
323     /* write an IPP response back to the network */
324     r->content_type = "application/ipp";
325
326 #ifndef APACHE2
327     ap_send_http_header(r);
328 #endif
329
330     status = ipp_write_message(write_data, r, response);
331     if (status != PAPI_OK)
332         _log_error(APLOG_MARK, APLOG_ERR, r,
333                    "write failed: %s\n", papiStatusString(status));
334 #ifdef DEBUG
335     fprintf(stderr, "write result: %s\n", papiStatusString(status));
336     fflush(stderr);
337 #endif
338
339     papiAttributeListFree(request);
340     papiAttributeListFree(response);
341
342 #ifndef APACHE2
343     ap_kill_timeout(r);
344     if (ap_rflush(r) < 0)
345         _log_error(APLOG_MARK, APLOG_ERR, r,
346                    "flush failed, response may not have been sent");
347 #endif
348
349     return (OK);
350 }

```

unchanged portion omitted

```
401 /*ARGSUSED0*/
402 static const char *
403 ipp_operation(cmd_parms *cmd, void *cfg, const char *op, const char *toggle)
402 ipp_operation(cmd_parms *cmd, void *cfg, char *op, char *toggle)
404 {
405     IPPListenerConfig *config = (IPPListenerConfig *)cfg;
406     papi_status_t status;
407
408     status = ipp_configure_operation(&config->operations, op, toggle);
409     switch (status) {
410     case PAPI_OK:
411         return (NULL);
412     case PAPI_BAD_ARGUMENT:
413         return (gettext("internal error (invalid argument)"));
414     default:
415         return (papiStatusString(status));
416     }
417
418     /* NOTREACHED */
419     /* return (gettext("contact your software vendor")); */
420 }
```

unchanged\_portion\_omitted\_

```
*****
23807 Wed Aug 8 20:59:37 2012
new/usr/src/man/man1m/logadm.1m
Bump Apache dependency to Apache 2
*****
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-\fIoptions\fR\fR] \fIlogname\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 \fIlogname\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 \fB&.0\fR (that is, \fIlogfile\fR\fB&.0\fR), the next most recent ends
36 with \fB&.1\fR (that is, \fIlogfile\fR\fB&.1\fR), and so forth. By default,
37 ten versions of old log files are kept (that is, \fIlogfile\fR\fB&.<0>\fR
38 through \fIlogfile\fR\fB&.9\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 \fIoptions\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 \fIlogname\fR
46 \fIoptions\fR
47 .sp
48 .ne 2
49 .na
50 \fB\fIlogname\fR\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 \fB\fIoptions\fR\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 \fIoptions\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 \fB*\fR and \fB?\fR, that are supported by
76 \fBcsh\fR(1).
77 .RE
78 .sp
79 .LP
80 Two options control when a log file is rotated. They are: \fB-s\fR size
81 \fB-p\fR period.
82 .sp
83 .LP
84 When using more than one of these options at a time, there is an implied
85 \fBband\fR between them. This means that all conditions must be met before the
86 log is rotated.
87 .sp
88 .LP
89 If neither of these two options are specified, the default conditions for
90 rotating a log file are: \fB\fR\fB-s\fR\fR \fB1b\fR\fR \fB-p\fR\fR \fB1w\fR\fR, which means
91 the log file is only rotated if the size is non-zero and if at least 1 week has
92 passed since the last time it was rotated.
93 .sp
94 .LP
95 By specifying \fB\fR\fB-p\fR\fR \fBnever\fR as a rotation condition, any other
96 rotation conditions are ignored and \fBlogadm\fR moves on to the expiration of
97 old log files. By specifying \fB\fR\fB-p\fR\fR \fBnow\fR as a rotation condition,
98 a log rotation is forced.
100 .sp
101 .LP
102 Unless specified by the \fB-o\fR, \fB-g\fR, or \fB-m\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: \fB-A\fR age \fB-C\fR
108 count \fB-S\fR size. These options expire the oldest log files until a
109 particular condition or conditions are met. For example, the combination
110 \fB\fR\fB-C\fR\fR \fB5\fR and the \fB\fR\fB-S\fR\fR \fB10m\fR options expires old log
111 files until there are no more than 5 of the \fBand\fR their combined disk usage
112 is no more than 10 megabytes. If none of these options are specified, the
113 default expiration is \fB\fR\fB-C\fR\fR \fB10\fR which keeps ten old log files. If
114 no files are to be expired, use \fB\fR\fB-C\fR\fR \fB0\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\fB-a\fR\fR \fIpost_command\fR\fR
124 .ad
125 .sp .6
126 .RS 4n
127 Execute the \fIpost_command\fR after renaming the log file. \fIpost_command\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
195 .sp
196 .ne 2
197 .na
198 \fB\fB-e\fR \fImail_addr\fR\fR
199 .ad
200 .sp .6
201 .RS 4n
202 Send error messages by email to \fImail_addr\fR.
203 .sp
204 As \fBlogadm\fR is typically run from \fBcron\fR(1M), error messages are
205 captured by \fBcron\fR and mailed to the owner of the \fBcrontab\fR.
206 .sp
207 This option is useful if you want the mail regarding error messages to go to
208 another address instead. If no errors are encountered, no mail message is
209 generated.
210 .RE

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

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

242 .sp
243 Create a new empty file with the \fBID\fR specified by \fIgroup\fR, instead of
244 preserving the group \fBID\fR of the log file.
245 .sp
246 Specify \fIgroup\fR by name or by numeric group \fBID\fR, as accepted by
247 \fBchgrp\fR(1).
248 .sp
249 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 \fBlogadm\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\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\fR
302 .ad
303 .sp .6
304 .RS 4n
305 Use \fIcmd\fR to rename the log file. If the keyword \fB$file\fR is specified,
306 it is expanded to the name of the log file. Similarly, the keyword \fB$logfile\fR
307 is expanded to the new name of the log file. The \fB$logfile\fR keyword is only
308 available with commands provided with the \fB-M\fR option. After the command
309 completes, the log file is replaced by the rotate file. The default \fIcmd\fR
310 is "\fB/bin/mv\fR \fB$file\fR\fB\fB$logfile\fR".
311 .RE

313 .sp
314 .ne 2
315 .na
316 \fB\fB-n\fR \fI\fR\fR
317 .ad
318 .sp .6
319 .RS 4n
320 Print the actions that the \fBlogadm\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, \fBlogadm\fR might not find files that need expiring,
328 but if run without the \fB-n\fR \fBlogadm\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 \fBlogadm\fR produces an error message if the log file is not found. With
343 \fB-N\fR, if the log file doesn't exist \fBlogadm\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\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\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\fR\fB-p\fR \fBnow\fR" forces log rotation. "\fB\fR\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\fR
381 .ad
382 .sp .6
383 .RS 4n
384 Used by \fBlogadm\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\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$file\fR is embedded in the specified command, it is expanded to the
414 name of the old log file just created by log rotation.
415 .sp
416 This option is useful for processing log file contents after rotating the log.
417 \fIcmd\fR is executed by passing it to \fB$fh\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\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\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\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\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\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\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$\fIfilename\fR\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\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\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.$n\fR.
603 Actual \fBrotate\fR of log files, where each version is shifted up until it
604 expires is done using the \fb$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 \fBlogadm\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 \fIlogname\fR exists in
647 the \fB/etc/logadm.conf\fR file and is syntactically correct. If \fIlogname\fR
648 is not specified, all entries in the configuration file are validated. If a
649 \fBlogname\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 \fBlogadm\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 \fBlogadm\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 Rotating 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 \fBlogadm\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 Rotating 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 Rotating /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 Rotating 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 Creating 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 \fBlogadm\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\fR Rotating 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/$basename.%Y-%m'\`e
831   -a '/usr/apache2/2.2/bin/apachectl graceful'\`e
832   '/var/apache2/2.2/logs/*{access,error}.log'
833   -t '/var/apache/old-logs/$basename.%Y-%m'\`e
834   -a '/usr/apache/bin/apachectl graceful'\`e
835   '/var/apache/logs/*{access,error}_log'

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 \fBlogadm\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 \fBfB/etc/logadm.conf\fR\fR
880 .ad
881 .sp .6
882 .RS 4n
883 configuration file for \fBlogadm\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 _Interface Stability Evolving
900 .TE

902 .SH SEE ALSO
903 .sp
904 .LP
905 \fBchgrp\fR(1), \fBchmod\fR(1), \fBchown\fR(1), \fBcsh\fR(1), \fBgzip\fR(1),
906 \fBcron\fR(1M), \fBctime\fR(3C), \fBstrftime\fR(3C), \fBlogadm.conf\fR(4),
907 \fBAttributes\fR(5)
908 .SH NOTES
909 .sp
910 .LP
911 When \fBlogadm\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, \fBlogadm\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.

```
new/usr/src/man/man4/mod_ipp.4
```

1

```
*****  
9243 Wed Aug 8 20:59:38 2012  
new/usr/src/man/man4/mod_ipp.4  
Bump Apache dependency to Apache 2  
*****  
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 \fB/usr/apache2/2.2/libexec/mod_ipp.so\fR  
14 \fB/usr/apache/libexec/mod_ipp.so\fR  
15 .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, \fByes\fR, \fBon\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 \fB\fBprint-job\fR\fR  
60 .ad
```

```
new/usr/src/man/man4/mod_ipp.4
```

2

```
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 \fB\fBprint-uri\fR\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 \fB\fBvalidate-job\fR\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 \fB\fBcreate-job\fR\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 \fB\fBget-jobs\fR\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 \fB\fBget-printer-attributes\fR\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 \fB\fBpause-printer\fR\fR  
125 .ad  
126 .RS 26n
```

**new/usr/src/man/man4/mod\_ipp.4**

3

```
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
```

**new/usr/src/man/man4/mod\_ipp.4**

4

```
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\fR Using 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/lp/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.

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

1

```
*****
2482 Wed Aug 8 20:59:39 2012
new/usr/src/pkg/manifests/print-lp-ipp-ipp-listener.mf
Bump Apache dependency to Apache 2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #

22 #
23 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
24 #

26 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=usr group=sys
42 dir path=usr/apache
43 dir path=usr/apache2/2.2
44 dir path=usr/apache2/2.2/libexec
45 dir path=var group=sys
46 dir path=var/lp group=lp mode=0775 owner=lp
47 dir path=var/lp/ipp-listener
48 file path=etc/apache2/2.2/httpd-standalone-ipp.conf \
49     original_name=SUNWipppl:etc/apache/httpd-standalone-ipp.conf preserve=true
50 file path=lib/svc/manifest/application/print/ipp-listener.xml group=sys \
51     mode=0444
52 file path=usr/apache2/2.2/libexec/mod_ipp.so mode=0555
53 file path=usr/apache/libexec/mod_ipp.so mode=0555
54 file path=var/lp/ipp-listener/index.html mode=0444
55 legacy pkg=SUNWipppl \
56     desc="Internet Printing Protocol(IPP) Apache configuration for service modul
      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=SUNWipppl \
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
```