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

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

# This file provides exceptions to the usual rules applied to ELF objects by
# check_rtime. All strings are Perl regular expressions that are compared to
# file paths. In addition to the standard Perl syntax, there is one extension:
#
#       MACH(dir)
#
# is expanded into a regular expression that matches the given
# directory, or a 64-bit subdirectory of the directory with the
# name of a 64-bit architecture. For example, MACH(lib) will match
# any of the following:
#
#       lib
#       lib/amd64
#       lib/sparcv9


# Directory hierarchies to skip completely
SKIP            ^usr/lib/libc/                  # optimized libc
SKIP            ^usr/lib/rcm/                   # 4426119
SKIP            ^usr/perl5/                     # alan’s taking care of these :-
SKIP            ^usr/src/                       # no objects in source code

# Individual files that we don’t examine
SKIP            ^boot/grub/bin/grub$
SKIP            ^usr/apache2/2.2/libexec/mod_ipp.so$    # Apache loadable module
SKIP            ^usr/apache/libexec/mod_ipp.so$ # Apache loadable module
# USIII specific extns. cause ldd noise on USII bld. m/c
SKIP            ^usr/lib/fps/sun4u/UltraSPARC.*/fptest$
SKIP            ^usr/MACH(lib)/lddstub$         # lddstub has no dependencies
SKIP            ^usr/MACH(lib)/libssagent\.so\.1$       # 4328854
SKIP            ^usr/lib/MACH(iconv)/geniconvtbl.so$    # 4384329

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

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

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

# Objects that are allowed to have an executable stack
EXEC_STACK      ^platform/.*/MACH(kernel)/unix$
EXEC_STACK      ^platform/.*/multiboot$

# Objects for which we allow relocations to the text segment
TEXTREL         ^platform/.*/MACH(kernel)/unix$

# Directories and files that are allowed to have no direct bound symbols
NODIRECT        ^platform/.*/MACH(kernel)/unix$
NODIRECT        ^usr/ucb
NODIRECT        ^usr/4lib/sbcp$

# Identify any files that should be skipped when building a crle(1)
# configuration file.  As the hwcap libraries can be loop-back mounted onto
# libc, these can confuse crle(1) because of their identical dev/inode.
NOCRLEALT       ^usr/lib/libc/libc_hwcap[1-3].so.1$

# Files that should contain debugging information.
STAB    ^platform/.*/MACH(kernel)/unix$

# Files that are allowed undefined references
UNDEF_REF       ^usr/lib/libnisdb\.so\.2$
UNDEF_REF       ^usr/snadm/lib/libsvm\.so\.1$

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

# libm.so.2 dependency
UNUSED_OBJ      unused object=.*MACH(libm)/libm_hwcap1\.so\.2

# Unused runpaths due to dlopen() use
UNUSED_RPATH    /usr/lib/fs/autofs.*\ from\ .automountd
UNUSED_RPATH    /etc/ppp/plugins.*\ from\ .*pppd
UNUSED_RPATH    /usr/lib/inet/ppp.*\ from\ .*pppd
UNUSED_RPATH    /usr/sfw/lib.*\ from\ .*libipsecutil\.so\.1
UNUSED_RPATH    /usr/platform/.*rsmlib.*\ from\ .*librsm\.so\.2
UNUSED_RPATH    \$ORIGIN.*\ from\ .*fcode.so
UNUSED_RPATH    /opt/VRTSvxvm/lib.*\ from\ .*libdiskmgt\.so\.1

# Unused runpaths in picl code
UNUSED_RPATH    /usr/platform/.*\ from\ .*/usr/platform
UNUSED_RPATH    /usr/lib/picl/.*\ from\ .*/usr/platform
UNUSED_RPATH    /usr/platform/.*\ from\ .*/usr/lib/picl

# Unused runpaths in non-OSNET objects we can’t change
UNUSED_RPATH    /usr/lib/mps.*\ from\ .*libnss3\.so
UNUSED_RPATH    /usr/lib/mps.*\ from\ .*libnssutil3\.so
UNUSED_RPATH    /usr/lib/mps.*\ from\ .*libsmime3\.so
UNUSED_RPATH    /usr/lib/mps.*\ from\ .*libssl3\.so
UNUSED_RPATH    /usr/sfw/lib.*\ from\ .*libdbus-1\.so\.3
UNUSED_RPATH    /usr/sfw/lib.*\ from\ .*libdbus-glib-1\.so\.2
UNUSED_RPATH    /usr/sfw/lib.*\ from\ .*libglib-2\.0\.so\.0
UNUSED_RPATH    /usr/X11/lib.*\ from\ .*libglib-2\.0\.so\.0
UNUSED_RPATH    /usr/sfw/lib.*\ from\ .*libgobject-2\.0\.so\.0
UNUSED_RPATH    /usr/X11/lib.*\ from\ .*libgobject-2\.0\.so\.0
UNUSED_RPATH    /usr/sfw/lib.*\ from\ .*libgthread-2\.0\.so\.0
UNUSED_RPATH    /usr/X11/lib.*\ from\ .*libgthread-2\.0\.so\.0
UNUSED_RPATH    /usr/sfw/lib.*\ from\ .*libcrypto\.so\.0\.9\.8
UNUSED_RPATH    /usr/sfw/lib.*\ from\ .*libnetsnmp\.so\.15
UNUSED_RPATH    /usr/sfw/lib.*\ from\ .*libgcc_s\.so\.1
UNUSED_RPATH    /usr/ccs/lib.*\ from\ .*libgcc_s\.so\.1
UNUSED_RPATH    /usr/lib.*\ from\ .*libgcc_s\.so\.1
UNUSED_RPATH    /usr/postgres/8.3/lib.*\ from\ .*libpq\.so\.5
UNUSED_RPATH    /usr/sfw/lib.*\ from\ .*libpq\.so\.5
UNUSED_RPATH    /usr/lib.*\ from\ .*/usr/lib/mps
UNUSED_RPATH    /usr/ccs/lib.*\ from\ .*/usr/lib/mps

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


# Unreferenced objects of non-OSnet objects we can't change
UNREF_OBJ       /lib.*\ of\ .*libcimapi\.so
UNREF_OBJ       /lib.*\ of\ .*libdbus-1\.so\.3
UNREF_OBJ       /lib.*\ of\ .*libdbus-glib-1\.so\.2
UNREF_OBJ       /lib.*\ of\ .*libglib-2.0\.so\.0
UNREF_OBJ       /lib.*\ of\ .*libgobject-2.0\.so\.0
UNREF_OBJ       /lib.*\ of\ .*libgthread-2\.0\.so\.0
UNREF_OBJ       /lib.*\ of\ .*libjvm\.so
UNREF_OBJ       /lib.*\ of\ .*libnetsnmp\.so\.15
UNREF_OBJ       /lib.*\ of\ .*libnetsnmpagent\.so\.15
UNREF_OBJ       /lib.*\ of\ .*libnetsnmpmibs\.so\.15
UNREF_OBJ       /lib.*\ of\ .*libnetsnmphelpers\.so\.15
UNREF_OBJ       /lib.*\ of\ .*libnspr4\.so
UNREF_OBJ       /lib.*\ of\ .*libpq\.so\.5
UNREF_OBJ       /lib.*\ of\ .*libsoftokn3\.so
UNREF_OBJ       /lib.*\ of\ .*libspmicommon\.so\.1
UNREF_OBJ       /lib.*\ of\ .*libspmocommon\.so\.1
UNREF_OBJ       /lib.*\ of\ .*libssl3\.so
UNREF_OBJ       /lib.*\ of\ .*libtspi\.so\.1
UNREF_OBJ       /lib.*\ of\ .*libxml2\.so\.2
UNREF_OBJ       /lib.*\ of\ .*libxslt\.so\.1
UNREF_OBJ       /lib.*\ of\ .*libpq\.so\.4
UNREF_OBJ       /lib.*\ of\ .*libpython2\.4\.so\.1\.0
UNREF_OBJ       /libgcc_s.*\ of\ .*libstdc\+\+\.so\.6

# Unreferenced object of objects we can't change for other reasons
UNREF_OBJ       /libmapmalloc\.so\.1;\ unused\ dependency\ of   # interposer
UNREF_OBJ       /libstdc\+\+\.so\.6;\ unused\ dependency\ of    # gcc build
UNREF_OBJ       /libm\.so\.2.*\ of\ .*libstdc\+\+\.so\.6        # gcc build
UNREF_OBJ       /lib.*\ of\ .*/lib/picl/plugins/                # picl
UNREF_OBJ       /lib.*\ of\ .*kcfd                              # interposer
UNREF_OBJ       /libpkcs11\.so\.1; .*\ of\ .*libkmf\.so\.1      # interposed
# Referenced by the Studio build, not the GCC build.  GCC eliminates the unused
# statics which have the dependence.
UNREF_OBJ       /libc\.so\.1.*\ of\ .*kldap\.so\.1


# Objects that used to contain system functionalty that has since
# migrated to libc. We preserve these libraries as pure filters for
# backward compatability but nothing needs to link to them.
OLDDEP          libaio\.so\.1                   # onnv build 44
OLDDEP          libdl\.so\.1                    # on10 build 49
OLDDEP          libdoor\.so\.1                  # onnv build 12
OLDDEP          libintl\.so\.1                  # on297 build 7
OLDDEP          libpthread\.so\.1               # on10 build 53
OLDDEP          librt\.so\.1                    # onnv build 44
OLDDEP          libsched\.so\.1                 # on10 build 36
OLDDEP          libthread\.so\.1                # on10 build 53
OLDDEP          libw\.so\.1                     # on297 build 7
```

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


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

```
**********************************************************
    3705 Wed Aug  8 20:59:27 2012
new/exception_lists/interface_check
Bump Apache dependency to Apache 2
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#


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

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


# Shared objects underneath these parts of the tree are taken to be plugins.
# Plugins are not required to have versioned file names, and are not required
# to be internally versioned.
#
PLUGIN          ^usr/apache2/2.2/libexec
PLUGIN          ^usr/apache/libexec
PLUGIN          ^usr/lib/devfsadm
PLUGIN          ^usr/lib/efcode/.*\.so$
PLUGIN          ^usr/lib/elfedit
PLUGIN          ^usr/lib/fm/fmd/plugins
PLUGIN          ^usr/lib/fm/fmd/schemes
PLUGIN          ^usr/lib/fm/topo/plugins
PLUGIN          ^usr/lib/fwflash
PLUGIN          ^usr/lib/iconv
PLUGIN          ^usr/lib/inet/ppp
PLUGIN          ^usr/lib/mdb
PLUGIN          ^usr/lib/pci
PLUGIN          ^usr/lib/picl/plugins
PLUGIN          ^usr/lib/python2.[46]
PLUGIN          ^usr/lib/rcm/modules
```

```
PLUGIN          ^usr/lib/scsi/plugins
PLUGIN          ^usr/lib/sysevent/modules
PLUGIN          ^usr/perl5/5\.[^\\]*/lib
PLUGIN          ^usr/platform
PLUGIN          ^usr/sadm/lib/wbem


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


# Objects that are not expected to contain versioning information.
# Note that PLUGIN objects are automatically exempt from this,
# so these directives are generally applied to non-plugin objects
NOVERDEF        ^usr/4lib/libc\.so\.
NOVERDEF        ^usr/MACH(lib)/0\@0\.so\.1$
NOVERDEF        ^usr/lib/MACH(abi)/apptrace\.so\.1$
NOVERDEF        ^usr/MACH(lib)/libfru.*\.so\.1$
NOVERDEF        ^usr/MACH(lib)/libkrb5\.so\.1$
NOVERDEF        ^usr/MACH(lib)/libzpool\.so\.1$
NOVERDEF        ^usr/MACH(lib)/madv\.so\.1$
NOVERDEF        ^usr/MACH(lib)/mpss\.so\.1$
NOVERDEF        ^usr/MACH(lib)/s10_brand\.so\.1$
NOVERDEF        ^usr/MACH(lib)/s10_npreload\.so\.1$
NOVERDEF        ^usr/MACH(lib)/sn1_brand\.so\.1$
NOVERDEF        ^usr/lib/fs/[^/]*/fstyp\.so\.1$
NOVERDEF        ^usr/lib/libmilter\.so\.1$
NOVERDEF        ^usr/lib/libwrap\.so\.1\.0$
NOVERDEF        ^usr/lib/locale/MACH(iso_8859_1)/iso_8859_1\.so\.3$
NOVERDEF        ^usr/lib/picl/plugins$
NOVERDEF        ^usr/sadm/admin/dhcpmgr/dhcpmgr\.so\.1$
NOVERDEF        ^usr/sadm/admin/printmgr/lib/libpmgr\.so\.1$


# Objects that are allowed to deviate from our standard version
# names.
NONSTD_VERNAME  ^usr/MACH(lib)/libtecla\.so\.1$


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


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

```
**********************************************************
   10285 Wed Aug  8 20:59:27 2012
new/usr/src/cmd/logadm/glob.c
Bump Apache dependency to Apache 2
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * logadm/glob.c -- globbing routines
 *
 * these routines support two kinds of globs.  first, the
 * usual kind of filename globbing, like:
 *
 *      *.c
 *      /var/log/syslog.?
 *      log[0-9]*file
 *      /var/apache2/2.2/logs/x*{access,error}_log
 *      /var/apache/logs/x*{access,error}_log
 *
 * this is basically the same syntax that csh supports for globs and
 * is provided by the routine glob_glob() which takes a filename and
 * returns a list of filenames that match the glob.
 *
 * the second type is something called a "reglob" which is a pathname
 * where the components are regular expressions as described in regex(3c).
 * some examples:
 *
 *      .*\.c
 *      /var/log/syslog\..
 *      log[0-9].*file
 *      /var/log/syslog\.([0-9]+)$0
 *
 * the last example uses the ()$n form to assign a numeric extension
 * on a filename to the "n" value kept by the fn routines with each
 * filename (see fn_setn() in fn.c).  logadm uses this mechanism to
 * correctly sort lognames when templates containing $n are used.
 *
 * the routine glob_reglob() is used to expand reglobs.  glob_glob()
 * is implemented by expanding the curly braces, converting the globs
 * to reglobs, and then passing the work to glob_reglob().
 *
 * finally, since expanding globs and reglobs requires doing a stat(2)
 * on the files, we store the resulting stat information in the filename
 * struct (see fn_setstat() in fn.c).
 *
```

```
 * the glob(3c) routines are not used here since they don't support
 * braces, and don't support the more powerful reglobs required by logadm.
 */

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

#include <stdio.h>
#include <libintl.h>
#include <stdlib.h>
#include <libgen.h>
#include <strings.h>
#include <sys/types.h>
#include <sys/param.h>
#include <sys/stat.h>
#include <dirent.h>
#include "err.h"
#include "fn.h"
#include "glob.h"

/* forward declarations for functions used internally by this module */
static struct fn_list *glob_debrace(struct fn *fnp);
static struct fn_list *glob_reglob_list(struct fn_list *fnlp);
static boolean_t glob_magic(struct fn *fnp);

/* expand curly braces (like file{one,two,three}name) */
static struct fn_list *
glob_debrace(struct fn *fnp)
{
        struct fn_list *ret = fn_list_new(NULL);
        struct fn_list *newret;
        char *sp = fn_s(fnp);
        char *left;
        char *right;
        char *comma;

        /* start with an empty string in the list */
        fn_list_adds(ret, "");

        /* while braces remain... */
        while (sp != NULL && (left = strchr(sp, '{')) != NULL)
                if ((right = strchr(left, '}')) == NULL) {
                        err(EF_FILE|EF_JMP, "Missing }");
                        fn_list_free(ret);
                        return (NULL);
                } else {
                        /* stuff before "left" is finished */
                        fn_list_appendrange(ret, sp, left);

                        /* stuff after "right" still need processing */
                        sp = right + 1;

                        if (left + 1 == right)
                                continue;       /* just an empty {} */

                        /* stuff between "left" and "right" is comma-sep list */
                        left++;
                        newret = fn_list_new(NULL);
                        while ((comma = strchr(left, ',')) != NULL) {
                                struct fn_list *dup = fn_list_dup(ret);

                                /* stuff from left to comma is one variant */
                                fn_list_appendrange(dup, left, comma);
                                fn_list_addfn_list(newret, dup);
                                left = comma + 1;
                        }
                        /* what's left is the last item in the list */
                        fn_list_appendrange(ret, left, right);
                        fn_list_addfn_list(newret, ret);
                        ret = newret;
                }

        /* anything remaining in "s" is finished */
        fn_list_appendrange(ret, sp, &sp[strlen(sp)]);
        return (ret);
}
```

______unchanged_portion_omitted_

```
**********************************************************
   34483 Wed Aug  8 20:59:28 2012
new/usr/src/cmd/logadm/main.c
Bump Apache dependency to Apache 2
**********************************************************
______unchanged_portion_omitted_
```

```c
/* absurdly long buffer lengths for holding user/group/mode strings */
#define TIMESTRMAX      100
#define MAXATTR         100

/* rotate a log file if necessary, returns true if ok to go on to expire step */
static boolean_t
rotatelog(struct fn *fnp, struct opts *opts)
{
        char *fname = fn_s(fnp);
        struct stat stbuf;
        char nowstr[TIMESTRMAX];
        struct fn *recentlog = fn_new(NULL);    /* for -R cmd */
        char ownerbuf[MAXATTR];
        char groupbuf[MAXATTR];
        char modebuf[MAXATTR];
        const char *owner;
        const char *group;
        const char *mode;

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

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

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

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

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

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

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

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

```c
                off_t when = opts_optarg_int(opts, "p");
                struct opts *cfopts;

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

                        /*
                         * last rotation is recorded as argument to -P,
                         * but if logname isn't the same as log file name
                         * then the timestamp would be recorded on a
                         * separate line in the conf file.  so if we
                         * haven't seen a -P already, we check to see if
                         * it is part of a specific entry for the log
                         * file name.  this handles the case where the
                         * logname is "apache", it supplies a log file
                         * name like "/var/apache2/2.2/logs/[a-z]*_log",
                         * which expands to multiple file names.  if one of
                         * the file names is "/var/apache2/2.2/logs/access_log"
                         * then the -P will be attached to a line with that
                         * name like "/var/apache/logs/[a-z]*_log",
                         * which expands to multiple file names.  if one
                         * of the file names is "/var/apache/logs/access_log"
                         * the the -P will be attached to a line with that
                         * logname in the conf file.
                         */
                        if (opts_count(opts, "P")) {
                                off_t last = opts_optarg_int(opts, "P");

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

                                /*
                                 * just checking this means this entry
                                 * is now "done" if we're going through
                                 * the entire conffile
                                 */
                                Donenames = lut_add(Donenames, fname, "1");

                                /* return if not enough time has passed */
                                if (Now - last < when)
                                        return (B_TRUE);
                        }
                }
        }

        if (Debug)
```

```
                (void) fprintf(stderr, "rotatelog: conditions met\n");
        if (opts_count(opts, "l")) {
                /* Change the time zone to local time zone */
                if (putenv("TZ="))
                        err(EF_SYS, "putenv TZ");
                tzset();
                Now = time(0);

                /* rename the log file */
                rotateto(fnp, opts, 0, recentlog, B_FALSE);

                /* Change the time zone to UTC */
                if (putenv("TZ=UTC"))
                        err(EF_SYS, "putenv TZ");
                tzset();
                Now = time(0);
        } else {
                /* rename the log file */
                rotateto(fnp, opts, 0, recentlog, B_FALSE);
        }

        /* determine owner, group, mode for empty log file */
        if (opts_count(opts, "o"))
                (void) strlcpy(ownerbuf, opts_optarg(opts, "o"), MAXATTR);
        else {
                (void) snprintf(ownerbuf, MAXATTR, "%ld", stbuf.st_uid);
        }
        owner = ownerbuf;
        if (opts_count(opts, "g"))
                group = opts_optarg(opts, "g");
        else {
                (void) snprintf(groupbuf, MAXATTR, "%ld", stbuf.st_gid);
                group = groupbuf;
        }
        (void) strlcat(ownerbuf, ":", MAXATTR - strlen(ownerbuf));
        (void) strlcat(ownerbuf, group, MAXATTR - strlen(ownerbuf));
        if (opts_count(opts, "m"))
                mode = opts_optarg(opts, "m");
        else {
                (void) snprintf(modebuf, MAXATTR,
                    "%03lo", stbuf.st_mode & 0777);
                mode = modebuf;
        }

        /* create the empty log file */
        docmd(opts, NULL, Touch, fname, NULL, NULL);
        docmd(opts, NULL, Chown, owner, fname, NULL);
        docmd(opts, NULL, Chmod, mode, fname, NULL);

        /* execute post-rotation command */
        if (opts_count(opts, "R")) {
                struct fn *rawcmd = fn_new(opts_optarg(opts, "R"));
                struct fn *cmd = fn_new(NULL);

                kw_init(recentlog, NULL);
                (void) kw_expand(rawcmd, cmd, 0, B_FALSE);
                docmd(opts, "-R cmd", Sh, "-c", fn_s(cmd), NULL);
                fn_free(rawcmd);
                fn_free(cmd);
        }
        fn_free(recentlog);

        /*
         * add "after" command to list of after commands.  we also record
         * the email address, if any, where the error output of the after
         * command should be sent.  if the after command is already on
```

```
         * our list, add the email addr to the list the email addrs for
         * that command (the after command will only be executed once,
         * so the error output gets mailed to every address we’ve come
         * across associated with this command).
         */
        if (opts_count(opts, "a")) {
                const char *cmd = opts_optarg(opts, "a");
                struct lut *addrs = (struct lut *)lut_lookup(Aftercmds, cmd);
                if (opts_count(opts, "e"))
                        addrs = lut_add(addrs, opts_optarg(opts, "e"), NULL);
                Aftercmds = lut_add(Aftercmds, opts_optarg(opts, "a"), addrs);
        }

        /* record the rotation date */
        (void) strftime(nowstr, sizeof (nowstr),
            "%a %b %e %T %Y", gmtime(&Now));
        if (opts_count(opts, "v") && fname != NULL)
                (void) out("#     recording rotation date %s for %s\n",
                    nowstr, fname);
        conf_set(fname, "P", STRDUP(nowstr));
        Donenames = lut_add(Donenames, fname, "1");
        return (B_TRUE);
}
```

______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_

#############
#############
#############    THE TESTS START AFTER HERE...
#############
#############

# common setup step -- create a testfile.conf
sub set_testconffile {
        my $fname = shift;
        $fname = 'testfile.conf' unless defined($fname);

        set_file($fname, <<'EOF');
#
# logadm.conf
#
# Default settings for system log file management.
# The -w option to logadm(1M) is the preferred way to write to this file,
# but if you do edit it by hand, use "logadm -V" to check it for errors.
#
# The format of lines in this file is:
#       <logname> <options>
# For each logname listed here, the default options to logadm
# are given.  Options given on the logadm command line override
# the defaults contained in this file.
#
# logadm typically runs early every morning via an entry in
# root's crontab (see crontab(1)).
#
/var/adm/messages -C 4 -P 'Thu Nov  1 16:56:42 2001' -a 'kill -HUP `cat /var/run
/var/cron/log -s 512k -t /var/cron/olog
/var/lp/logs/lpsched -C 2 -N -t '$file.$N'
#
# The entry below is used by turnacct(1M)
#
/var/adm/pacct -C 0 -a '/usr/lib/acct/accton pacct' -g adm -m 664 -o adm -p neve
apache -C 24 -a '/usr/apache2/2.2/bin/apachectl graceful' -p 1m -t '/var/apache2
apache -C 24 -a '/usr/apache/bin/apachectl graceful' -p 1m -t '/var/apache/old-l
/var/log/syslog -C 8 -P 'Thu Nov  1 09:16:38 2001' -a 'kill -HUP `cat /var/run/s
/var/apache2/2.2/logs/access_log -P 'Thu Nov  1 08:27:56 2001'
/var/apache2/2.2/logs/error_log -P 'Thu Nov  1 08:27:56 2001'
/var/apache2/2.2/logs/suexec_log -P 'Thu Nov  1 08:27:56 2001'
/var/apache2/2.2/logs/mod_jserv.log -P 'Thu Nov  1 08:27:56 2001'
/var/apache2/2.2/logs/jserv.log -P 'Thu Nov  1 08:27:56 2001'
/var/apache/logs/access_log -P 'Thu Nov  1 08:27:56 2001'
/var/apache/logs/error_log -P 'Thu Nov  1 08:27:56 2001'
/var/apache/logs/suexec_log -P 'Thu Nov  1 08:27:56 2001'
/var/apache/logs/mod_jserv.log -P 'Thu Nov  1 08:27:56 2001'
/var/apache/logs/jserv.log -P 'Thu Nov  1 08:27:56 2001'
EOF
}
______unchanged_portion_omitted_

###########################################################################
#
#       logadmV1 -- test of "logadm -V"
#
###########################################################################
sub logadmV1 {
        set_testconffile;
```

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

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

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

```
**********************************************************
    4199 Wed Aug  8 20:59:30 2012
new/usr/src/cmd/lp/cmd/lpsched/print-svc
Bump Apache dependency to Apache 2
**********************************************************
#!/sbin/sh
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#

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

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

case "$1" in
’start’)

isopts=‘/usr/sbin/svccfg <<-EOF
        select $SMF_FMRI
        listpg cmd_opts

        EOF‘

if [ "$isopts" ] ; then

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

        num_notifiers=‘/bin/svcprop -p cmd_opts/num_notifiers $SMF_FMRI‘

        if [ "$num_notifiers" != "" ] ; then
                OPTS="$OPTS -n $num_notifiers"
        fi

        num_filters=‘/bin/svcprop -p cmd_opts/num_filters $SMF_FMRI‘

        if [ "$num_filters" != "" ]  ; then
                OPTS="$OPTS -f $num_filters"
        fi

        fd_limit=‘/bin/svcprop -p cmd_opts/fd_limit $SMF_FMRI‘

        if [ "$fd_limit" != "" ]  ; then
```

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

        reserved_fds=‘/bin/svcprop -p cmd_opts/reserved_fds $SMF_FMRI‘

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

# clear out cmd_opts property group

        svccfg <<-EOF
        select $SMF_FMRI
        delpg cmd_opts

        EOF

else

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

        num_notifiers=‘/bin/svcprop -p lpsched/num_notifiers $SERVICE‘
        if [ "$num_notifiers" != "" ] && [ "$num_notifiers" != "0" ] ; then
                OPTS="$OPTS -n $num_notifiers"
        fi

        num_filters=‘/bin/svcprop -p lpsched/num_filters $SERVICE‘
        if [ "$num_filters" != "" ] && [ "$num_filters" != "0" ] ; then
                OPTS="$OPTS -f $num_filters"
        fi

        fd_limit=‘/bin/svcprop -p lpsched/fd_limit $SERVICE‘
        if [ "$fd_limit" != "" ]  && [ "$fd_limit" != "0" ]; then
                OPTS="$OPTS -p $fd_limit"
        fi

        reserved_fds=‘/bin/svcprop -p lpsched/reserved_fds $SERVICE‘
        if [ "$reserved_fds" != "" ] && [ "$reserved_fds" != "0" ] ; then
                OPTS="$OPTS -r $reserved_fds"
        fi
fi

# set temporary or permanent properties from OPTS

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

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

        ;;

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

        /usr/lib/lp/local/lpshut
        ;;

’ipfilter’)
        FMRI=$2
        IPP_FMRI="svc:/application/print/ipp-listener:default"
        RFC1179_FMRI="svc:/application/print/rfc1179:default"
        IPP_CONF=/etc/apache2/2.2/httpd-standalone-ipp.conf
        IPP_CONF=/etc/apache/httpd-standalone-ipp.conf
        ip="any"

        policy=‘get_policy $FMRI‘
        file=‘fmri_to_file $RFC1179_FMRI $IPF_SUFFIX‘
        echo "# $RFC1179_FMRI" >$file
        service_is_enabled ${RFC1179_FMRI}
        if [ $? -eq 0 ]; then
                rfc_name=‘svcprop -p inetd/name ${RFC1179_FMRI} 2>/dev/null‘
                rfc_proto=‘svcprop -p inetd/proto ${RFC1179_FMRI} 2>/dev/null |
                    sed ’s/6/ /’‘
                rfc_port=‘$SERVINFO -p -t -s $rfc_name‘
                generate_rules $FMRI $policy $rfc_proto $ip $rfc_port $file
        fi

        file=‘fmri_to_file $IPP_FMRI $IPF_SUFFIX‘
        echo "# $IPP_FMRI" >$file
        service_is_enabled ${IPP_FMRI}
        if [ $? -eq 0 ]; then
                #
                # If Listen directives are used, it’s possibie to listen on
                # more than one ports. Process the Port directives only when Lis
                # directives don’t exist.
                #
                ipp_ports=‘grep ’^[ \t]*[^# ]*Listen’ ${IPP_CONF} | awk ’{print

                if [ -z "$ipp_ports" ]; then
                        ipp_ports=‘grep ’^[ \t]*[^# ]*Port’ ${IPP_CONF} | \
                            awk ’{print $2}’ | tail -1‘
                fi

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

        ;;

*)
        echo "Usage: $0 { start | stop }"
        exit 1
        ;;
esac
exit $SMF_EXIT_OK
```

```
**********************************************************
   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
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2012 Nexenta Systems, Inc.  All rights reserved.
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 */

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

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

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <netinet/in.h>
#include <assert.h>
#include <errno.h>
#include <syslog.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/systeminfo.h>

#include <papi.h>
#include <ipp-listener.h>
#include <uri.h>

typedef papi_status_t (ipp_handler_t)(papi_service_t svc,
                                        papi_attribute_t **request,
                                        papi_attribute_t ***response,
                                        ipp_reader_t iread, void *fd);

/*
 * protocol request handlers are inserted below.  The handler must be
 * declared extern immediately below this comment and then an entry
 * must be inserted in the "handlers" table a little further down.
 */
extern ipp_handler_t ipp_print_job;
extern ipp_handler_t ipp_validate_job;
```

```
extern ipp_handler_t ipp_create_job;
extern ipp_handler_t ipp_get_printer_attributes;
extern ipp_handler_t ipp_get_jobs;
extern ipp_handler_t ipp_pause_printer;
extern ipp_handler_t ipp_resume_printer;
extern ipp_handler_t ipp_disable_printer;
extern ipp_handler_t ipp_enable_printer;
extern ipp_handler_t ipp_purge_jobs;
extern ipp_handler_t ipp_send_document;
extern ipp_handler_t ipp_cancel_job;
extern ipp_handler_t ipp_get_job_attributes;
extern ipp_handler_t ipp_release_job;
extern ipp_handler_t ipp_hold_job;
extern ipp_handler_t ipp_restart_job;
extern ipp_handler_t ipp_set_job_attributes;
extern ipp_handler_t ipp_set_printer_attributes;
extern ipp_handler_t cups_get_default;
extern ipp_handler_t cups_get_printers;
extern ipp_handler_t cups_get_classes;
extern ipp_handler_t cups_accept_jobs;
extern ipp_handler_t cups_reject_jobs;
extern ipp_handler_t cups_move_job;

/* ARGSUSED0 */
static papi_status_t
default_handler(papi_service_t svc, papi_attribute_t **request,
                papi_attribute_t ***response, ipp_reader_t iread, void *fd)
{
        int result = (int)PAPI_INTERNAL_ERROR;

        if (response != NULL)
                (void) papiAttributeListGetInteger(*response, NULL,
                                        "status-code", &result);

        return ((papi_status_t)result);
}
______unchanged_portion_omitted_

static int
ipp_operation_name_to_index(const char *name)
ipp_operation_name_to_index(char *name)
{
        int i;

        for (i = 0; handlers[i].name != NULL; i++)
                if (strcasecmp(name, handlers[i].name) == 0)
                        return (i);

        return (-1);
}
______unchanged_portion_omitted_

static char
type_to_boolean(const char *type)
type_to_boolean(char *type)
{
        char result = PAPI_FALSE;

        if ((strcasecmp(type, "true") == 0) ||
            (strcasecmp(type, "yes") == 0) ||
            (strcasecmp(type, "on") == 0) ||
            (strcasecmp(type, "enable") == 0))
                result = PAPI_TRUE;

        return (result);
}
______unchanged_portion_omitted_
```

```
papi_status_t
ipp_configure_operation(papi_attribute_t ***list, const char *operation, const c
ipp_configure_operation(papi_attribute_t ***list, char *operation, char *type)
{
        papi_status_t result = PAPI_OPERATION_NOT_SUPPORTED;
        char boolean = PAPI_FALSE;

        if ((list == NULL) || (operation == NULL) || (type == NULL))
                return (PAPI_BAD_ARGUMENT);

        boolean = type_to_boolean(type);

        if (strcasecmp(operation, "all") == 0) {
                result = ipp_configure_all_operations(list, boolean);
        } else if (strcasecmp(operation, "required") == 0) {
                result = ipp_configure_required_operations(list, boolean);
        } else if (ipp_operation_name_to_index(operation) != -1) {
                result = papiAttributeListAddBoolean(list, PAPI_ATTR_REPLACE,
                                                        operation, boolean);
        }

        return (result);
}
______unchanged_portion_omitted_
```

```
**********************************************************
    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
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

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

#ifndef _IPP_LISTENER_H
#define _IPP_LISTENER_H

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

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

#ifdef  __cplusplus
extern "C" {
#endif

#include <ipp.h>

/* exported functions */
extern papi_status_t ipp_configure_operation(papi_attribute_t ***list,
                                                const char *operation,
                                                const char *type);
                                                char *operation, char *type);
extern papi_status_t ipp_process_request(papi_attribute_t **request,
                                        papi_attribute_t ***response,
                                        ipp_reader_t iread, void *fd);

/* shared internal functions */
extern char *ipp_svc_status_mesg(papi_service_t svc, papi_status_t status);
extern char *destination_from_printer_uri(char *);
extern void get_printer_id(papi_attribute_t **attributes, char **printer,
                        int *id);
extern void ipp_operations_supported(papi_attribute_t ***list,
                        papi_attribute_t **request);
extern void get_string_list(papi_attribute_t **attributes, char *name,
                        char ***values);
extern void add_default_attributes(papi_attribute_t ***attributes);
extern void papi_to_ipp_printer_group(papi_attribute_t ***response,
```

```
                        papi_attribute_t **request, int flags,
                        papi_printer_t p);
extern void papi_to_ipp_job_group(papi_attribute_t ***response,
                        papi_attribute_t **request, int flags, papi_job_t j);
extern void massage_response(papi_attribute_t **request,
                        papi_attribute_t **response);

#ifdef  __cplusplus
}
______unchanged_portion_omitted_
```

```
**********************************************************
   25631 Wed Aug  8 20:59:33 2012
new/usr/src/lib/print/libpapi-common/common/attribute.c
Bump Apache dependency to Apache 2
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2012 Nexenta Systems, Inc.  All rights reserved.
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 */

/* $Id: attribute.c 157 2006-04-26 15:07:55Z ktou $ */

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

/*LINTLIBRARY*/

#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <ctype.h>
#include <alloca.h>
#include <papi.h>
#include <regex.h>

#define MAX_PAGES 32767
/*
 * Assuming the maximum number of pages in
 * a document to be 32767
 */

static void papiAttributeFree(papi_attribute_t *attribute);

static void
papiAttributeValueFree(papi_attribute_value_type_t type,
                    papi_attribute_value_t *value)
{
        if (value != NULL) {
                switch (type) {
                case PAPI_STRING:
                        if (value->string != NULL)
                                free(value->string);
                        break;
```

```
                case PAPI_COLLECTION:
                        if (value->collection != NULL) {
                                int i;

                                for (i = 0; value->collection[i] != NULL; i++)
                                        papiAttributeFree(value->collection[i]);

                                free(value->collection);
                        }
                        break;
                default: /* don’t need to free anything extra */
                        break;
                }

                free(value);
        }
}
_____unchanged_portion_omitted_

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

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

        return (result);
}
_____unchanged_portion_omitted_

papi_status_t
papiAttributeListAddValue(papi_attribute_t ***list, int flgs,
                const char *name, papi_attribute_value_type_t type,
                char *name, papi_attribute_value_type_t type,
                papi_attribute_value_t *value)
{
        papi_status_t result;
        int flags = flgs;
        papi_attribute_t *attribute = NULL;
        papi_attribute_value_t **values = NULL;

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

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

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

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

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

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

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

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

        /*
         * if we are replacing, clear any existing values, but don’t free
         * until after we have replaced the values, in case we are replacing
         * a collection with a relocated version of the original collection.
         */
        if (((flags & PAPI_ATTR_REPLACE) != 0) && (attribute->values != NULL)) {
                values = attribute->values;
                attribute->values = NULL;
        }

        attribute->type = type;

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

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

        return (result);
}
```

______***unchanged_portion_omitted_***

```
papi_status_t
papiAttributeListAddBoolean(papi_attribute_t ***list, int flags,
                        const char *name, char boolean)
                        char *name, char boolean)
{
        papi_attribute_value_t v;

        v.boolean = boolean;
        return (papiAttributeListAddValue(list, flags, name, PAPI_BOOLEAN, &v));
}
```

______***unchanged_portion_omitted_***

```
papi_attribute_t *
papiAttributeListFind(papi_attribute_t **list, const char *name)
papiAttributeListFind(papi_attribute_t **list, char *name)
{
        int i;
        if ((list == NULL) || (name == NULL))
                return (NULL);

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

        return (NULL);
}
```

______***unchanged_portion_omitted_***

```
**********************************************************
   15768 Wed Aug  8 20:59:34 2012
new/usr/src/lib/print/libpapi-common/common/papi.h
Bump Apache dependency to Apache 2
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2012 Nexenta Systems, Inc.  All rights reserved.
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 */

#ifndef _PAPI_H
#define _PAPI_H

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

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

#include <sys/types.h>
#include <time.h>
#include <stdio.h>

#ifdef __cplusplus
extern "C" {
#endif

/*
 * Types
 */

/*      service related types   */
typedef void *papi_service_t;
typedef void *papi_printer_t;
typedef void *papi_job_t;
typedef void *papi_stream_t;

typedef enum {
        PAPI_ENCRYPT_IF_REQUESTED,      /* Encrypt if requested (TLS upgrade) */
        PAPI_ENCRYPT_NEVER,             /* Never encrypt */
        PAPI_ENCRYPT_REQUIRED,          /* Encryption required (TLS upgrade) */
        PAPI_ENCRYPT_ALWAYS             /* Always encrypt (SSL) */
} papi_encryption_t;
______unchanged_portion_omitted_
```

```
/*
 * Functions
 */

/*      Service related         */
extern papi_status_t papiServiceCreate(papi_service_t *handle,
                                        char *service_name, char *user_name,
                                        char *password,
                                        int (*authCB)(papi_service_t svc,
                                                        void *app_data),
                                        papi_encryption_t encryption,
                                        void *app_data);
extern void papiServiceDestroy(papi_service_t handle);
extern papi_status_t papiServiceSetUserName(papi_service_t handle,
                                        char *user_name);
extern papi_status_t papiServiceSetPassword(papi_service_t handle,
                                        char *password);
extern papi_status_t papiServiceSetEncryption(papi_service_t handle,
                                        papi_encryption_t encryption);
extern papi_status_t papiServiceSetAuthCB(papi_service_t handle,
                                        int (*authCB)(papi_service_t s,
                                                        void *app_data));
extern papi_status_t papiServiceSetAppData(papi_service_t handle,
                                        void *app_data);
extern char *papiServiceGetServiceName(papi_service_t handle);
extern char *papiServiceGetUserName(papi_service_t handle);
extern char *papiServiceGetPassword(papi_service_t handle);
extern papi_encryption_t papiServiceGetEncryption(papi_service_t handle);
extern void *papiServiceGetAppData(papi_service_t handle);
extern papi_attribute_t **papiServiceGetAttributeList(papi_service_t handle);
extern char *papiServiceGetStatusMessage(papi_service_t handle);

/*      Attribute related       */
extern papi_status_t papiAttributeListAddValue(papi_attribute_t ***attrs,
                                        int flags, const char *name,
                                        int flags, char *name,
                                        papi_attribute_value_type_t type,
                                        papi_attribute_value_t *value);
extern papi_status_t papiAttributeListAddString(papi_attribute_t ***attrs,
                                        int flags, char *name, char *string);
extern papi_status_t papiAttributeListAddInteger(papi_attribute_t ***attrs,
                                        int flags, char *name, int integer);
extern papi_status_t papiAttributeListAddBoolean(papi_attribute_t ***attrs,
                                        int flags, const char *name, char boolea
                                        int flags, char *name, char boolean);
extern papi_status_t papiAttributeListAddRange(papi_attribute_t ***attrs,
                                        int flags, char *name,
                                        int lower, int upper);
extern papi_status_t papiAttributeListAddResolution(papi_attribute_t ***attrs,
                                        int flags, char *name,
                                        int xres, int yres,
                                        papi_resolution_unit_t units);
extern papi_status_t papiAttributeListAddDatetime(papi_attribute_t ***attrs,
                                        int flags, char *name, time_t datetime);
extern papi_status_t papiAttributeListAddCollection(papi_attribute_t ***attrs,
                                        int flags, char *name,
                                        papi_attribute_t **collection);
extern papi_status_t papiAttributeListAddMetadata(papi_attribute_t ***attrs,
                                        int flags, char *name,
                                        papi_metadata_t metadata);
extern papi_status_t papiAttributeListDelete(papi_attribute_t ***attributes,
                                        char *name);
extern papi_status_t papiAttributeListGetValue(papi_attribute_t **list,
                                        void **iterator, char *name,
                                        papi_attribute_value_type_t type,
                                        papi_attribute_value_t **value);
```

```
extern papi_status_t papiAttributeListGetString(papi_attribute_t **list,
                                        void **iterator, char *name,
                                        char **vptr);
extern papi_status_t papiAttributeListGetInteger(papi_attribute_t **list,
                                        void **iterator, char *name, int *vptr);
extern papi_status_t papiAttributeListGetBoolean(papi_attribute_t **list,
                                        void **iterator, char *name,
                                        char *vptr);
extern papi_status_t papiAttributeListGetRange(papi_attribute_t **list,
                                        void **iterator, char *name,
                                        int *min, int *max);
extern papi_status_t papiAttributeListGetResolution(papi_attribute_t **list,
                                        void **iterator, char *name,
                                        int *x, int *y,
                                        papi_resolution_unit_t *units);
extern papi_status_t papiAttributeListGetDatetime(papi_attribute_t **list,
                                        void **iterator, char *name,
                                        time_t *dt);
extern papi_status_t papiAttributeListGetCollection(papi_attribute_t **list,
                                        void **iterator, char *name,
                                        papi_attribute_t ***collection);
extern papi_status_t papiAttributeListGetMetadata(papi_attribute_t **list,
                                        void **iterator, char *name,
                                        papi_metadata_t *vptr);
extern papi_attribute_t *papiAttributeListFind(papi_attribute_t **list,
                                        const char *name);
                                        char *name);
extern papi_attribute_t *papiAttributeListGetNext(papi_attribute_t **list,
                                        void **iterator);
extern void papiAttributeListFree(papi_attribute_t **attributes);

extern papi_status_t papiAttributeListFromString(papi_attribute_t ***attrs,
                                        int flags, char *string);
extern papi_status_t papiAttributeListToString(papi_attribute_t **attrs,
                                        char *delim,
                                        char *buffer, size_t buflen);
extern void papiAttributeListPrint(FILE *fp, papi_attribute_t **list,
                                        char *prefix_fmt, ...);

/*      Printer related          */
extern papi_status_t papiPrintersList(papi_service_t handle,
                                        char **requested_attrs,
                                        papi_filter_t *filter,
                                        papi_printer_t **printers);
extern papi_status_t papiPrinterQuery(papi_service_t handle, char *name,
                                        char **requested_attrs,
                                        papi_attribute_t **job_attributes,
                                        papi_printer_t *printer);
extern papi_status_t papiPrinterAdd(papi_service_t handle, char *name,
                                        papi_attribute_t **attributes,
                                        papi_printer_t *printer);
extern papi_status_t papiPrinterModify(papi_service_t handle, char *name,
                                        papi_attribute_t **attributes,
                                        papi_printer_t *printer);
extern papi_status_t papiPrinterRemove(papi_service_t handle, char *name);
extern papi_status_t papiPrinterDisable(papi_service_t handle, char *name,
                                        char *message);
extern papi_status_t papiPrinterEnable(papi_service_t handle, char *name);
extern papi_status_t papiPrinterPause(papi_service_t handle, char *name,
                                        char *message);
extern papi_status_t papiPrinterResume(papi_service_t handle, char *name);
extern papi_status_t papiPrinterPurgeJobs(papi_service_t handle,
                                        char *name, papi_job_t **jobs);
extern papi_status_t papiPrinterListJobs(papi_service_t handle,
                                        char *name, char **requested_attrs,
                                        int type_mask, int max_num_jobs,
```

```
                                        papi_job_t **jobs);
extern papi_attribute_t **papiPrinterGetAttributeList(papi_printer_t printer);
extern void papiPrinterFree(papi_printer_t printer);
extern void papiPrinterListFree(papi_printer_t *printers);

/*      Job related             */
extern papi_status_t papiJobSubmit(papi_service_t handle, char *printer,
                                        papi_attribute_t **job_attributes,
                                        papi_job_ticket_t *job_ticket,
                                        char **files, papi_job_t *job);
extern papi_status_t papiJobSubmitByReference(papi_service_t handle,
                                        char *printer,
                                        papi_attribute_t **job_attributes,
                                        papi_job_ticket_t *job_ticket,
                                        char **files, papi_job_t *job);
extern papi_status_t papiJobValidate(papi_service_t handle, char *printer,
                                        papi_attribute_t **job_attributes,
                                        papi_job_ticket_t *job_ticket,
                                        char **files, papi_job_t *job);
extern papi_status_t papiJobStreamOpen(papi_service_t handle,
                                        char *printer,
                                        papi_attribute_t **job_attributes,
                                        papi_job_ticket_t *job_ticket,
                                        papi_stream_t *stream);
extern papi_status_t papiJobStreamWrite(papi_service_t handle,
                                        papi_stream_t stream,
                                        void *buffer, size_t buflen);
extern papi_status_t papiJobStreamClose(papi_service_t handle,
                                        papi_stream_t stream,
                                        papi_job_t *job);
extern papi_status_t papiJobQuery(papi_service_t handle, char *printer,
                                        int32_t job_id, char **requested_attrs,
                                        papi_job_t *job);
extern papi_status_t papiJobModify(papi_service_t handle, char *printer,
                                        int32_t job_id,
                                        papi_attribute_t **attributes,
                                        papi_job_t *job);
extern papi_status_t papiJobMove(papi_service_t handle, char *printer,
                                        int32_t job_id, char *destination);
extern papi_status_t papiJobCancel(papi_service_t handle, char *printer,
                                        int32_t job_id);
extern papi_status_t papiJobHold(papi_service_t handle, char *printer,
                                        int32_t job_id);
extern papi_status_t papiJobRelease(papi_service_t handle, char *printer,
                                        int32_t job_id);
extern papi_status_t papiJobRestart(papi_service_t handle, char *printer,
                                        int32_t job_id);
extern papi_status_t papiJobPromote(papi_service_t handle, char *printer,
                                        int32_t job_id);
extern papi_attribute_t **papiJobGetAttributeList(papi_job_t printer);
extern char *papiJobGetPrinterName(papi_job_t printer);
extern int32_t papiJobGetId(papi_job_t printer);
extern papi_job_ticket_t *papiJobGetJobTicket(papi_job_t printer);
extern void papiJobFree(papi_job_t job);
extern void papiJobListFree(papi_job_t *jobs);

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

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

extern char *papiStatusString(papi_status_t status);

/*
 * Internal functions that aren't in the API, but are shared across
 * protocol support implementations(psms) and the tightly bound
 * listener library.  Do not use these in your applications.
 */
extern void list_append();
extern void list_concatenate();
extern void list_remove();
extern void copy_attributes(papi_attribute_t ***result,
                                papi_attribute_t **list);
extern void split_and_copy_attributes(char **list,
                                papi_attribute_t **attributes,
                                papi_attribute_t ***in,
                                papi_attribute_t ***out);

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

extern int is_localhost(char *hostname);

#ifdef __cplusplus
}
______unchanged_portion_omitted_
```

```
**********************************************************
    2673 Wed Aug  8 20:59:35 2012
new/usr/src/lib/print/mod_ipp/Makefile
Bump Apache dependency to Apache 2
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2012 Nexenta Systems, Inc.  All rights reserved.
# Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

LIBRARY =       mod_ipp.a
VERS =
OBJECTS =       mod_ipp.o

include ../../Makefile.lib
include ../../Makefile.rootfs

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

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

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

LIBS =                  $(DYNLIB)

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

APR_INC.cmd =   apr-1-config --includes
APU_INC.cmd =   apu-1-config --includes
CFLAGS +=       $(CCVERBOSE) -erroff=E_IMPLICIT_DECL_FUNC_RETURN_INT

CFLAGS +=       $(CCVERBOSE)
CPPFLAGS +=     -I../libipp-listener/common
CPPFLAGS +=     -I../libipp-core/common
CPPFLAGS +=     -I/usr/apache2/2.2/include
CPPFLAGS +=     $(APR_INC.cmd:sh)
CPPFLAGS +=     $(APU_INC.cmd:sh)
CPPFLAGS +=     -I/usr/apache/include
```

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

MAPFILES =      mapfile

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

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

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

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

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

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

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

$(ROOTDIRS):
        $(INS.dir)

.KEEP_STATE:

all:    $(LIBS)

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

install_h:

lint:

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

```
**********************************************************
   14473 Wed Aug  8 20:59:36 2012
new/usr/src/lib/print/mod_ipp/mod_ipp.c
Bump Apache dependency to Apache 2
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

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

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

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

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

#include "ap_config.h"

#include <stdio.h>
#include <time.h>
#include <sys/time.h>
#include <values.h>
#include <libintl.h>
#include <alloca.h>

#include "httpd.h"
#include "http_config.h"
#include "http_core.h"
#include "http_protocol.h"
#include "http_log.h"
#include "http_main.h"
#include "papi.h"
#ifndef APACHE_RELEASE  /* appears to only exist in Apache 1.X */
#define APACHE2
#define ap_table_get apr_table_get
#include "apr_compat.h"
#endif

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

```
#ifndef APACHE2
module MODULE_VAR_EXPORT ipp_module;
#else
module AP_MODULE_DECLARE_DATA ipp_module;
#endif

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

typedef struct {
        int conformance;
        char *default_user;
        char *default_svc;
        papi_attribute_t **operations;
} IPPListenerConfig;
______unchanged_portion_omitted_

void _log_rerror(const char *file, int line, int level, request_rec *r,
        const char *fmt, ...)
{
        va_list args;
        size_t size;
        char *message = alloca(BUFSIZ);

        va_start(args, fmt);
        /*
         * fill in the message.  If the buffer is too small, allocate
         * one that is large enough and fill it in.
         */
        if ((size = vsnprintf(message, BUFSIZ, fmt, args)) >= BUFSIZ)
                if ((message = alloca(size)) != NULL)
                        vsnprintf(message, size, fmt, args);
        va_end(args);

#ifdef APACHE2
        ap_log_rerror(file, line, level, APR_SUCCESS, r, message);
        ap_log_rerror(file, line, level, NULL, r, message);
#else
        ap_log_rerror(file, line, level, r, message);
#endif
}

static int
ipp_handler(request_rec *r)
{
        papi_attribute_t **request = NULL, **response = NULL;
        IPPListenerConfig *config;
        papi_status_t status;
        int ret;

        /* Really, IPP is all POST requests */
        if (r->method_number != M_POST)
                return (DECLINED);

#ifndef APACHE2
        /*
         * An IPP request must have a MIME type of "application/ipp"
         * (RFC-2910, Section 4, page 19).  If it doesn’t match this
         * MIME type, we should decline the request and let someone else
         * try and handle it.
         */
        if (r->headers_in != NULL) {
                char *mime_type = (char *)ap_table_get(r->headers_in,
                                                        "Content-Type");

                if ((mime_type == NULL) ||
                    (strcasecmp(mime_type, "application/ipp") != 0))
                        return (DECLINED);
        }
#endif
        /* CHUNKED_DECHUNK might not work right for IPP? */
        if ((ret = ap_setup_client_block(r, REQUEST_CHUNKED_DECHUNK)) != OK)
                return (ret);

        if (!ap_should_client_block(r))
                return (HTTP_INTERNAL_SERVER_ERROR);

#ifndef APACHE2
        ap_soft_timeout("ipp_module: read/reply request ", r);
#endif
        /* read the IPP request off the network */
        status = ipp_read_message(read_data, r, &request, IPP_TYPE_REQUEST);

        if (status != PAPI_OK)
                _log_rerror(APLOG_MARK, APLOG_ERR, r,
                        "read failed: %s\n", papiStatusString(status));
#ifdef DEBUG
        papiAttributeListPrint(stderr, request, "request (%d)  ", getpid());
#endif

        (void) papiAttributeListAddString(&request, PAPI_ATTR_EXCL,
                "originating-host", (char *)
#ifdef APACHE2
                ap_get_remote_host
                        (r->connection, r->per_dir_config, REMOTE_NAME, NULL));
#else
                ap_get_remote_host
                        (r->connection, r->per_dir_config, REMOTE_NAME));
#endif

        (void) papiAttributeListAddInteger(&request, PAPI_ATTR_EXCL,
                                "uri-port", ap_get_server_port(r));
        if (r->headers_in != NULL) {
                char *host = (char *)ap_table_get(r->headers_in, "Host");

                if ((host == NULL) || (host[0] == '\0'))
                        host = (char *)ap_get_server_name(r);

                (void) papiAttributeListAddString(&request, PAPI_ATTR_EXCL,
                                "uri-host", host);
        }
        (void) papiAttributeListAddString(&request, PAPI_ATTR_EXCL,
                                "uri-path", r->uri);

        config = ap_get_module_config(r->per_dir_config, &ipp_module);
        if (config != NULL) {
                (void) papiAttributeListAddInteger(&request, PAPI_ATTR_EXCL,
                                "conformance", config->conformance);
                (void) papiAttributeListAddCollection(&request, PAPI_ATTR_EXCL,
                                "operations", config->operations);
                if (config->default_user != NULL)
                        (void) papiAttributeListAddString(&request,
                                                PAPI_ATTR_EXCL, "default-user",
                                                config->default_user);
                if (config->default_svc != NULL)
                        (void) papiAttributeListAddString(&request,
```

```
                                        PAPI_ATTR_EXCL, "default-service",
                                        config->default_svc);
        }

#ifndef APACHE2
        /*
         * For Trusted Solaris, pass the fd number of the socket connection
         * to the backend so the it can be forwarded to the backend print
         * service to retrieve the sensativity label off of a multi-level
         * port.
         */
        (void) papiAttributeListAddInteger(&request, PAPI_ATTR_EXCL,
                        "peer-socket", ap_bfileno(r->connection->client, B_RD));
#endif

        /* process the request */
        status = ipp_process_request(request, &response, read_data, r);
        if (status != PAPI_OK) {
                errno = 0;
                _log_rerror(APLOG_MARK, APLOG_ERR, r,
                        "request failed: %s\n", papiStatusString(status));
                discard_data(r);
        }
#ifdef DEBUG
        fprintf(stderr, "processing result: %s\n", papiStatusString(status));
        papiAttributeListPrint(stderr, response, "response (%d)  ", getpid());
#endif

        /*
         * If the client is using chunking and we have not yet received the
         * final "0" sized chunk, we need to discard any data that may
         * remain in the post request.
         */
        if ((r->read_chunked != 0) &&
            (ap_table_get(r->headers_in, "Content-Length") == NULL))
                discard_data(r);

        /* write an IPP response back to the network */
        r->content_type = "application/ipp";

#ifndef APACHE2
        ap_send_http_header(r);
#endif

        status = ipp_write_message(write_data, r, response);
        if (status != PAPI_OK)
                _log_rerror(APLOG_MARK, APLOG_ERR, r,
                        "write failed: %s\n", papiStatusString(status));
#ifdef DEBUG
        fprintf(stderr, "write result: %s\n", papiStatusString(status));
        fflush(stderr);
#endif

        papiAttributeListFree(request);
        papiAttributeListFree(response);

#ifndef APACHE2
        ap_kill_timeout(r);
        if (ap_rflush(r) < 0)
                _log_rerror(APLOG_MARK, APLOG_ERR, r,
                        "flush failed, response may not have been sent");
#endif

        return (OK);
}
```

______unchanged_portion_omitted_

```
/*ARGSUSED0*/
static const char *
ipp_operation(cmd_parms *cmd, void *cfg, const char *op, const char *toggle)
ipp_operation(cmd_parms *cmd, void *cfg, char *op, char *toggle)
{
        IPPListenerConfig *config = (IPPListenerConfig *)cfg;
        papi_status_t status;

        status = ipp_configure_operation(&config->operations, op, toggle);
        switch (status) {
        case PAPI_OK:
                return (NULL);
        case PAPI_BAD_ARGUMENT:
                return (gettext("internal error (invalid argument)"));
        default:
                return (papiStatusString(status));
        }

        /* NOTREACHED */
        /* return (gettext("contact your software vendor")); */
}
```

______***unchanged_portion_omitted_***

****
   23807 Wed Aug 8 20:59:37 2012
new/usr/src/man/man1m/logadm.1m
Bump Apache dependency to Apache 2
****

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

.LP
.nf
\fBlogadm\fR [\fB-\fIoptions\fR\fR] \fIlogname\fR...
.fi

.SH DESCRIPTION
.sp
.LP
\fBlogadm\fR is a general log rotation tool that is suitable for running from
\fBcron\fR(1M).
.sp
.LP
Without arguments, \fBlogadm\fR reads the \fB/etc/logadm.conf\fR file, and for
every entry found in that file checks the corresponding log file to see if it
should be rotated. Typically this check is done each morning by an entry in the
root’s \fBcrontab\fR.
.sp
.LP
If the \fIlogname\fR argument is specified, \fBlogadm\fR renames the
corresponding log file by adding a suffix so that the most recent log file ends
with \fB\&.0\fR (that is, \fIlogfile\fR\fB\&.0\fR), the next most recent ends
with \fB\&.1\fR (that is, \fIlogfile\fR\fB\&.1\fR), and so forth. By default,
ten versions of old log files are kept (that is, \fIlogfile\fR\fB\&.0\fR
through \fIlogfile\fR\fB\&.9\fR). At the point when what would be the eleventh
file is logged, \fBlogadm\fR automatically deletes the oldest version to keep
the count of files at ten.
.sp
.LP
\fBlogadm\fR takes a number of \fIoptions\fR. You can specify these options on
the command line or in the \fB/etc/logadm.conf\fR file. The \fBlogadm\fR
command searches \fB/etc/logadm.conf\fR for lines of the form \fIlogname\fR
\fIoptions\fR
.sp
.ne 2
.na
\fB\fIlogname\fR\fR
.ad
.sp .6
.RS 4n
Identifies an entry in \fB/etc/logadm.conf\fR. This can be a name or the
pathname of the log file. If you specify a log file, rather than a name, for
this field, it must be a fully qualified pathname.
.RE

.sp
.ne 2
.na
```

```
\fB\fIoptions\fR\fR
.ad
.sp .6
.RS 4n
Identifies command line options exactly as they would be entered on the command
line. This allows commonly used log rotation policies to be stored in the
\fB/etc/logadm.conf\fR file. See \fBEXAMPLES\fR.
.sp
If \fIoptions\fR are specified both in \fB/etc/logadm.conf\fR and on the
command line, those in the \fB/etc/logadm.conf\fR file are applied first.
Therefore, the command line options override those in \fB/etc/logadm.conf\fR.
.sp
Log file names specified in \fB/etc/logadm.conf\fR may contain filename
substitution characters such as \fB*\fR and \fB?\fR, that are supported by
\fBcsh\fR(1).
.RE

.sp
.LP
Two options control when a log file is rotated. They are: \fB-s\fR size
\fB-p\fR period.
.sp
.LP
When using more than one of these options at a time, there is an implied
\fBand\fR between them. This means that all conditions must be met before the
log is rotated.
.sp
.LP
If neither of these two options are specified, the default conditions for
rotating a log file are: \fB\fR\fB-s\fR \fB1b\fR \fB-p\fR \fB1w\fR, which means
the log file is only rotated if the size is non-zero and if at least 1 week has
passed since the last time it was rotated.
.sp
.LP
By specifying \fB\fR\fB-p\fR \fBnever\fR as a rotation condition, any other
rotation conditions are ignored and \fBlogadm\fR moves on to the expiration of
old log files. By specifying \fB\fR\fB-p\fR \fBnow\fR as a rotation condition,
a log rotation is forced.
.sp
.LP
Unless specified by the \fB-o\fR, \fB-g\fR, or \fB-m\fR options, \fBlogadm\fR
replaces the log file (after renaming it) by creating an empty file whose
owner, group ID, and permissions match the original file.
.sp
.LP
Three options control when old log files are expired: \fB-A\fR age \fB-C\fR
count \fB-S\fR size. These options expire the oldest log files until a
particular condition or conditions are met. For example, the combination
\fB\fR\fB-C\fR \fB5\fR and the \fB\fR\fB-S\fR \fB10m\fR options expires old log
files until there are no more than 5 of the \fBand\fR their combined disk usage
is no more than 10 megabytes. If none of these options are specified, the
default expiration is \fB\fR\fB-C\fR \fB10\fR which keeps ten old log files. If
no files are to be expired, use \fB\fR\fB-C\fR \fB0\fR to prevent expiration by
default.
.SH OPTIONS
.sp
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-a\fR \fIpost_command\fR\fR
.ad
.sp .6
.RS 4n
Execute the \fIpost_command\fR after renaming the log file. \fIpost_command\fR
```

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

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

.sp
.ne 2
.na
\fB\fB-b\fR \fIpre_command\fR\fR
.ad
.sp .6
.RS 4n
Execute \fIpre_command\fR before renaming the log file. \fIpre_command\fR is
passed to \fBsh\fR \fB-c\fR.
.sp
Specify \fIpre_command\fR as a valid shell command. Use quotes to protect
spaces or shell metacharacters in the \fIpre_command\fR.
.sp
This option can be used to stop a daemon that is writing to the file. When
rotating multiple logs with one \fBlogadm\fR command, \fIpre_command\fR is
executed only once before all the logs are rotated, not once per rotated log.
.RE

.sp
.ne 2
.na
\fB\fB-c\fR\fR
.ad
.sp .6
.RS 4n
Rotate the log file by copying it and truncating the original logfile to zero
length, rather than renaming the file.
.RE

.sp
.ne 2
.na
\fB\fB-C\fR \fIcount\fR\fR
.ad
.sp .6
.RS 4n
Delete the oldest versions until there are not more than \fIcount\fR files
left.
.sp
If no expire options (\fB-A\fR, \fB-C\fR, or \fB-S\fR) are specified,
\fB\fR\fB-C\fR \fB10\fR is the default. To prevent the default expire rule from
being added automatically, specify \fB\fR\fB-C\fR \fB0\fR .
```

```
.RE

.sp
.ne 2
.na
\fB\fB-e\fR \fImail_addr\fR\fR
.ad
.sp .6
.RS 4n
Send error messages by email to \fImail_addr\fR.
.sp
As \fBlogadm\fR is typically run from \fBcron\fR(1M), error messages are
captured by \fBcron\fR and mailed to the owner of the \fBcrontab\fR.
.sp
This option is useful if you want the mail regarding error messages to go to
another address instead. If no errors are encountered, no mail message is
generated.
.RE

.sp
.ne 2
.na
\fB\fB-E\fR \fIcmd\fR\fR
.ad
.sp .6
.RS 4n
Execute \fIcmd\fR to expire the file, rather than deleting the old log file to
expire it.
.sp
\fIcmd\fR is passed it to \fBsh\fR \fB-c\fR. The file is considered expired
after \fIcmd\fR completes. If the old log file is not removed or renamed by the
\fIcmd\fR, \fBlogadm\fR considers it for expiration the next time that it runs
on the specified log file. If present, the keyword \fB$file\fR is expanded in
the specified \fIcmd\fRto the name of the file being expired.
.sp
This option is useful for tasks such as mailing old log files to
administrators, or copying old log files to long term storage.
.RE

.sp
.ne 2
.na
\fB\fB-f\fR \fIconf_file\fR\fR
.ad
.sp .6
.RS 4n
Use \fIconf_file\fR instead of \fB/etc/logadm.conf\fR.
.sp
This option allows non-root users to keep their own \fBlogadm\fR configuration
files.
.RE

.sp
.ne 2
.na
\fB\fB-g\fR \fIgroup\fR\fR
.ad
.sp .6
.RS 4n
Create a new empty file with the \fBID\fR specified by \fIgroup\fR, instead of
preserving the group \fBID\fR of the log file.
.sp
Specify \fIgroup\fR by name or by numeric group \fBID\fR, as accepted by
\fBchgrp\fR(1).
.sp
This option requires the ability to change file group ownership using the
```

```
\fBchgrp\fR(1) command.
.RE

.sp
.ne 2
.na
\fB\fB-h\fR\fR
.ad
.sp .6
.RS 4n
Print a help message that describes \fBlogadm\fR’s options.
.RE

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

.sp
.ne 2
.na
\fB\fB-m\fR \fImode\fR\fR
.ad
.sp .6
.RS 4n
Create a new empty file with the mode specified by \fImode\fR, instead of
preserving the mode of the log file.
.sp
Specify \fImode\fR in any form that is accepted by the \fBchmod\fR(1) command.
.RE

.sp
.ne 2
.na
\fB\fB-M\fR \fIcmd\fR\fR
.ad
.sp .6
.RS 4n
Use \fIcmd\fR to rename the log file. If the keyword \fB$file\fR is specified,
it is expanded to the name of the log file. Similarly, the keyword \fB$nfile\fR
is expanded to the new name of the log file. The \fB$nfile\fR keyword is only
available with commands provided with the \fB-M\fR option. After the command
completes, the log file is replaced by the rotate file. The default \fIcmd\fR
is "\fB/bin/mv\fR \fB$file\fR\fB\fR\fB$nfile\fR".
.RE

.sp
.ne 2
.na
\fB\fB-n\fR \fI\fR\fR
.ad
.sp .6
.RS 4n
Print the actions that the \fBlogadm\fR command will perform without actually
performing them.
.sp
This option is useful for checking arguments before making any changes to the
system.
.sp
```

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

.sp
.ne 2
.na
\fB\fB-N\fR\fR
.ad
.sp .6
.RS 4n
Prevent an error message if the specified logfile does not exist. Normally,
\fBlogadm\fR produces an error message if the log file is not found. With
\fB-N\fR, if the log file doesn’t exist \fBlogadm\fR moves on to the expire
rules (if any) and then to the next log file (if any), without creating the
empty replacement log file.
.RE

.sp
.ne 2
.na
\fB\fB-o\fR \fIowner\fR\fR
.ad
.sp .6
.RS 4n
Create the new empty file with \fIowner\fR, instead of preserving the owner of
the log file.
.sp
Specify \fIowner\fR in any form that is accepted by the \fBchown\fR(1) command.
.RE

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

.sp
.ne 2
.na
\fB\fB-P\fR \fItimestamp\fR\fR
.ad
.sp .6
.RS 4n
Used by \fBlogadm\fR to record the last time the log was rotated in
\fB/etc/logadm.conf\fR.
.sp
This option uses \fItimestamp\fR to determine if the log rotation period has
passed. The format of \fItimestamp\fR matches the format generated by
\fBctime\fR(3C), with quotes around it to protect embedded spaces.
\fItimestamp\fR is always recorded in the Coordinated Universal Time (UTC)
timezone.
```

```
.RE

.sp
.ne 2
.na
\fB\fB-r\fR\fR
.ad
.sp .6
.RS 4n
Remove any entries corresponding to the specified \fIlogname\fR from the
\fB/etc/logadm.conf\fR.
.RE

.sp
.ne 2
.na
\fB\fB-R\fR \fIcmd\fR\fR
.ad
.sp .6
.RS 4n
Run the \fIcmd\fR when an old log file is created by a log rotation. If the
keyword \fB$file\fR is embedded in the specified command, it is expanded to the
name of the old log file just created by log rotation.
.sp
This option is useful for processing log file contents after rotating the log.
\fIcmd\fR is executed by passing it to \fBsh\fR \fB-c\fR. When rotating
multiple logs with one logadm command, the command supplied with \fB-R\fR is
executed once every time a log is rotated. This is useful for post-processing a
log file (that is, sorting it, removing uninteresting lines, etc.). The
\fB-a\fR option is a better choice for restarting daemons after log rotation.
.RE

.sp
.ne 2
.na
\fB\fB-s\fR \fIsize\fR\fR
.ad
.sp .6
.RS 4n
Rotate the log file only if its size is greater than or equal to \fIsize\fR.
.sp
Specify \fIsize\fR as a number followed by the letter \fBb\fR for bytes,
\fBk\fR for kilobytes, \fBm\fR for megabytes, or \fBg\fR for gigabytes.
.RE

.sp
.ne 2
.na
\fB\fB-S\fR \fIsize\fR\fR
.ad
.sp .6
.RS 4n
Delete the oldest versions until the total disk space used by the old log files
is less than the specified size.
.sp
Specify \fIsize\fR as a number followed by the letter \fBb\fR for bytes,
\fBk\fR for kilobytes, \fBm\fR for megabytes, or \fBg\fR for gigabytes.
.RE

.sp
.ne 2
.na
\fB\fB-t\fR \fItemplate\fR\fR
.ad
.sp .6
.RS 4n
```

```
Specify the template to use when renaming log files.
.sp
\fItemplate\fR can be a simple name, such as \fB/var/adm/oldfile\fR, or it can
contain special keywords which are expanded by \fBlogadm\fR and are in the form
\fB$\fR\fIword\fR. Allowed sequences are:
.sp
.ne 2
.na
\fB$\fIbasename\fR\fR
.ad
.sp .6
.RS 4n
The log file name, without the directory name
.RE

.sp
.ne 2
.na
\fB$\fIdirname\fR\fR
.ad
.sp .6
.RS 4n
The directory of the file to be rotated
.RE

.sp
.ne 2
.na
\fB$\fIdomain\fR\fR
.ad
.sp .6
.RS 4n
Expands to the output of \fBdomainname\fR
.RE

.sp
.ne 2
.na
\fB$\fIfile\fR\fR
.ad
.sp .6
.RS 4n
The full path name of the file to be rotated
.RE

.sp
.ne 2
.na
\fB$\fIisa\fR\fR
.ad
.sp .6
.RS 4n
Expands to the output of \fBuname\fR \fB-p\fR
.RE

.sp
.ne 2
.na
\fB$\fImachine\fR\fR
.ad
.sp .6
.RS 4n
Expands to the output of \fBuname\fR \fB-m\fR
.RE

.sp
```

```
.ne 2
.na
\fB$\fIn\fR\fR
.ad
.sp .6
.RS 4n
The version number, \fB0\fR is most recent, \fB1\fR is next most recent, and so
forth
.RE

.sp
.ne 2
.na
\fB$\fIN\fR\fR
.ad
.sp .6
.RS 4n
The same as $\fIn\fR, but starts at \fB1\fR instead of zero
.RE

.sp
.ne 2
.na
\fB$\fInodename\fR\fR
.ad
.sp .6
.RS 4n
Expands to the output of \fBuname\fR \fB-n\fR
.RE

.sp
.ne 2
.na
\fB$\fIplatform\fR\fR
.ad
.sp .6
.RS 4n
Expands to the output of \fBuname\fR \fB-i\fR
.RE

.sp
.ne 2
.na
\fB$\fIrelease\fR\fR
.ad
.sp .6
.RS 4n
Expands to the output of \fBuname\fR \fB-r\fR
.RE

.sp
.ne 2
.na
\fB$\fIsecs\fR\fR
.ad
.sp .6
.RS 4n
The number of seconds since \fB00:00:00 UTC, January 1,1970\fR
.RE

.sp
.ne 2
.na
\fB$\fIzonename\fR\fR
.ad
.sp .6
```

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

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

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

.sp
.ne 2
.na
\fB\fB-v\fR\fR
.ad
.sp .6
.RS 4n
Print information about the actions being executed in verbose mode.
.RE

.sp
.ne 2
.na
\fB\fB-V\fR\fR
.ad
.sp .6
.RS 4n
Validate the configuration file.
.sp
This option validates that an entry for the specified \fIlogname\fR exists in
the \fB/etc/logadm.conf\fR file and is syntactically correct. If \fIlogname\fR
is not specified, all entries in the configuration file are validated. If a
\fBlogname\fR argument is specified, the command validates the syntax of that
entry. If the entry is found, it is printed and the exit value of the command
is true. Otherwise the exit value is false.
.RE

.sp
.ne 2
```

```
.na
\fB\fB-w\fR \fIentryname\fR\fR
.ad
.sp .6
.RS 4n
Write an entry into the config file (that is, \fB/etc/logadm.conf\fR) that
corresponds to the current command line arguments. If an entry already existed
for the specified \fIentryname\fR, it is removed first. This is the preferred
method for updating \fB/etc/logadm.conf\fR, because it prevents syntax errors.
The \fIentryname\fR is an argument to an invocation of \fBlogadm\fR.
\fIentryname\fR might be chosen as something easy to remember or it can be the
pathname of the log file. If a pathname, rather than a name is used, it must be
a fully qualified pathname.
.sp
If no log file name is provided on a \fBlogadm\fR command line, the entry name
is assumed to be the same as the log file name. For example, the following two
lines achieve the same thing, keeping two copies of rotated log files:
.sp
.in +2
.nf
% logadm -C2 -w mylog /my/really/long/log/file/name
% logadm -C2 -w /my/really/long/log/file/name
.fi
.in -2
.sp

.RE

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

.SH OPERANDS
.sp
.LP
The following operands are supported:
.sp
.ne 2
.na
\fB\fIlogname\fR\fR
.ad
.sp .6
.RS 4n
Identifies the name of the entry in \fB/etc/logadm.conf\fR. If the log file
name is specified in the \fIlogname\fR field, it is assumed that \fIlogname\fR
is the same as the actual log file name.
.RE

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

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

.sp
.LP
Tell \fBlogadm\fR to copy the file and truncate it.

.sp
.in +2
.nf
 % logadm -c /var/adm/exacct/proc
.fi
.in -2
.sp

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

.sp
.in +2
.nf
% logadm -C8 -t'/var/oldlogs/syslog.$n' /var/log/syslog
.fi
.in -2
.sp

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

.sp
.in +2
.nf
/var/adm/sulog -A 30d
.fi
.in -2
.sp

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

.sp
.in +2
.nf
/var/adm/sulog -S 100m
.fi
.in -2
.sp

.LP
\fBExample 5 \fRCreating an Entry that Stores the Logfile Name
.sp
.LP
This example creates an entry storing the log file name and the fact that we
```

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

.sp
.in +2
.nf
% logadm -w locallog /usr/local/logfile -C20 -p never
.fi
.in -2
.sp

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

.sp
.in +2
.nf
 % logadm -p now locallog
.fi
.in -2
.sp

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

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

.sp
.in +2
.nf
 % logadm -w apache -p 1m -C 24\e
     -t '/var/apache2/2.2/old-logs/$basename.%Y-%m'\e
     -a '/usr/apache2/2.2/bin/apachectl graceful'\e
     '/var/apache2/2.2/logs/*{access,error}_log'
     -t '/var/apache/old-logs/$basename.%Y-%m'\e
     -a '/usr/apache/bin/apachectl graceful'\e
     '/var/apache/logs/*{access,error}_log'
.fi
.in -2
.sp

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

.sp
.in +2
.nf
% logadm -p now apache
.fi
.in -2
.sp
```

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

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

.sp
.in +2
.nf
logadm -w apache /var/apache2/2.2/logs/*_log
logadm -w apache /var/apache/logs/*_log
.fi
.in -2
.sp

.SH FILES
.sp
.ne 2
.na
\fB\fB/etc/logadm.conf\fR\fR
.ad
.sp .6
.RS 4n
configuration file for \fBlogadm\fR command
.RE

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE	ATTRIBUTE VALUE
_
Interface Stability	Evolving
.TE

.SH SEE ALSO
.sp
.LP
\fBchgrp\fR(1), \fBchmod\fR(1), \fBchown\fR(1), \fBcsh\fR(1), \fBgzip\fR(1),
\fBcron\fR(1M), \fBctime\fR(3C), \fBstrftime\fR(3C), \fBlogadm.conf\fR(4),
\fBattributes\fR(5)
.SH NOTES
.sp
.LP
When \fBlogadm\fR applies expire conditions (supplied by the \fB-A\fR,
\fB-C\fR, and \fB-S\fR options), it deletes files, the oldest first, until the
conditions are satisfied. If the template used for naming the old logs
contained \fB$n\fR or \fB$N\fR, \fBlogadm\fR picks the highest value of
\fB$n\fR or \fB$N\fR found in the old log file names first. If the template
```

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

```
**********************************************************
    9243 Wed Aug  8 20:59:38 2012
new/usr/src/man/man4/mod_ipp.4
Bump Apache dependency to Apache 2
**********************************************************
'\" te
.\" Copyright (c) 2003, Sun Microsystems, Inc.
.\" All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH MOD_IPP 4 "Nov 26, 2003"
.SH NAME
mod_ipp \- Embedded Internet Print Protocol (IPP) listener for the Apache HTTP
server
.SH SYNOPSIS
.LP
.nf
\fB/usr/apache2/2.2/libexec/mod_ipp.so\fR
\fB/usr/apache/libexec/mod_ipp.so\fR
.fi

.SH DESCRIPTION
.sp
.LP
The \fBmod_ipp\fR module implements RFCs 2910 and 2911 to provide an IPP
handling service for the Apache HTTP server. When loaded on the Apache server,
\fBmod_ipp\fR processes all HTTP requests with MIME types of
\fBapplication/ipp\fR. The \fBmod_ipp\fR module also processes additional
configuration directives to enable or disable portions of the protocol support.
.SS "Using Configuration Directives"
.sp
.LP
The following is a list of configuration directives that apply to the Apache
IPP Listening service:
.RS +4
.TP
.ie t \(bu
.el o
ipp-conformance (\fIautomatic\fR|\fI1.0\fR|\fI1.1\fR)
.RE
.RS +4
.TP
.ie t \(bu
.el o
ipp-operation (\fIoperation\fR) (\fBenable\fR|\fBdisable\fR)
.RS +4
.TP
.ie t \(bu
.el o
\fBenable\fR|\fBdisable\fR
.sp
The values \fBtrue\fR, \fByes\fR, \fBon\fR, \fBenable\fR are considered to be
synonymous and will enable support for the named operation. All other values
will disable support for the named operation.
.RE
.RE
.SS "Operations"
.sp
.LP
The following is a list of IPP handling service operations:
.sp
.ne 2
.na
\fB\fBprint-job\fR\fR
.ad
```

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

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

.sp
.ne 2
.na
\fB\fBvalidate-job\fR\fR
.ad
.RS 26n
This is a required IPP operation that allows client systems to simulate the
submission of a print job to verify that the server is capable of handling the
job. This operation is supported by \fBmod_ipp\fR.
.RE

.sp
.ne 2
.na
\fB\fBcreate-job\fR\fR
.ad
.RS 26n
This is an optional IPP operation that allows client systems to submit a print
job. The operation is used with the \fBsend-document\fR and \fBsend-uri\fR
operations.
.RE

.sp
.ne 2
.na
\fB\fBget-jobs\fR\fR
.ad
.RS 26n
This is a required IPP operation that allows client systems to retrieve a list
of print jobs from the print service.
.RE

.sp
.ne 2
.na
\fB\fBget-printer-attributes\fR\fR
.ad
.RS 26n
This is a required IPP operation that allows client systems to retrieve
attributes from the print service that describes the named printer object.
.RE

.sp
.ne 2
.na
\fB\fBpause-printer\fR\fR
.ad
.RS 26n
```

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

.sp
.ne 2
.na
\fB\fBresume-printer\fR\fR
.ad
.RS 26n
This is an optional IPP operation that allows client systems to resume job
processing on the named print queue.
.RE

.sp
.ne 2
.na
\fB\fBpurge-jobs\fR\fR
.ad
.RS 26n
This is an optional IPP operation that allows client systems to cancel all
print jobs on the named print queue.
.RE

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

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

.sp
.ne 2
.na
\fB\fBcancel-job\fR\fR
.ad
.RS 26n
This is a required IPP operation that allows client systems to cancel print
jobs.
.RE

.sp
.ne 2
.na
\fB\fBget-job-attributes\fR\fR
.ad
.RS 26n
This is a required IPP operation that allows client systems to retrieve
attributes that describe a print job from the print service.
.RE
```

```
.sp
.ne 2
.na
\fB\fBhold-job\fR\fR
.ad
.RS 26n
This is an optional IPP operation that allows client systems to hold print
jobs.
.RE

.sp
.ne 2
.na
\fB\fBrelease-job\fR\fR
.ad
.RS 26n
This is an optional IPP operation that allows client systems to release print
jobs.
.RE

.sp
.ne 2
.na
\fB\fBrestart-job\fR\fR
.ad
.RS 26n
This is an optional IPP operation that allows client systems to restart print
jobs.
.RE

.sp
.ne 2
.na
\fB\fBall\fR\fR
.ad
.RS 26n
This is a place holder for enabling or disabling support for all IPP operations
implemented by the \fBmod_ipp\fR Apache module.
.RE

.sp
.ne 2
.na
\fB\fBrequired\fR\fR
.ad
.RS 26n
This is a place holder for enabling or disabling support for the required IPP
operations implemented by the \fBmod_ipp\fR Apache module.
.RE

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

.sp
.in +2
.nf
ServerType standalone
ServerRoot "/usr/apache2/2.2"
ServerRoot "/usr/apache"
PidFile /var/run/httpd-standalone-ipp.pid
ErrorLog /var/lp/logs/ipp-errors
```

```
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

MinSpareServers 1
MaxSpareServers 3
StartServers 1
MaxClients 150
MaxRequestsPerChild 0

LoadModule ipp_module libexec/mod_ipp.so

ClearModuleList
AddModule mod_ipp.c
AddModule mod_so.c

Port 631

User lp
Group lp

ServerAdmin lp@localhost
DefaultType application/ipp

<IFModule mod_app>
        <Location />
                ipp-operation all on
        </Location>
</IFModule mod_app>
.fi
.in -2

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

.sp
.in +2
.nf
<IFModule mod_app>
        <Location />
                ipp-operation all offn
                ipp-operation required on
        </Location>
</IFModule mod_app>
.fi
.in -2

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) or descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Interface Stability     External
.TE
```

```
.SH SEE ALSO
.sp
.LP
\fBman\fR(1), \fBcatman\fR(1M), \fBattributes\fR(5)
.sp
.LP
Herriot, R., Ed., Butler, S., Moore, P., Turner, R., Wenn, J. \fIRFC 2910,
Internet Printing Protocol/1.1: Encoding and Transport\fR. Network Working
Group. September 2000.
.sp
.LP
Hastings, T., Ed., Herriot, R., deBry, R., Isaacson, S., Powell, P. \fIRFC
2911, Internet Printing Protocol/1.1: Model and Semantics\fR. Network Working
Group. September 2000.
.sp
.LP
http://www.apache.org
.SH NOTES
.sp
.LP
Configuration file directives are processed in the order listed in the config
file. The default behavior is to enable support for all operations implemented
in the \fBmod_ipp\fR Apache module.
.sp
.LP
Since the Apache IPP listening service implements some capabilities that are
more of operator features, it may not be desirable to enable all IPP operations
without requiring user authentication on the Apache listening service.
.sp
.LP
The following is an example of a more reasonable configuration for Apache IPP
servers without user authentication enabled:
.sp
.in +2
.nf
ipp-operations  all      disabled
ipp-operations  required enabled
.fi
.in -2

.sp
.LP
The printers and jobs available under this service can be accessed using URIs
of the following form:
.sp
.in +2
.nf
printer:
                 http://server[:port]/printers/{queue}
                 ipp://server[:port]/printers/{queue}
job:
                 http://server[:port]/printers/{queue}/{job-id}
                 ipp://server[:port]/printers/{queue}/{job-id}
.fi
.in -2

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

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

```
**********************************************************
    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
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

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

set name=pkg.fmri value=pkg:/print/lp/ipp/ipp-listener@$(PKGVERS)
set name=pkg.description \
    value="Internet Printing Protocol(IPP) Apache module for listening service"
set name=pkg.summary value="Internet Printing Protocol(IPP) listener module"
set name=info.classification \
    value=org.opensolaris.category.2008:System/Printing
set name=variant.arch value=$(ARCH)
dir path=etc group=sys
dir path=etc/apache2
dir path=etc/apache2/2.2
dir path=etc/apache
dir path=lib
dir path=lib/svc
dir path=lib/svc/manifest group=sys
dir path=lib/svc/manifest/application group=sys
dir path=lib/svc/manifest/application/print group=sys
dir path=usr group=sys
dir path=usr/apache2
dir path=usr/apache2/2.2
dir path=usr/apache2/2.2/libexec
dir path=usr/apache
dir path=usr/apache/libexec
dir path=var group=sys
dir path=var/lp group=lp mode=0775 owner=lp
dir path=var/lp/ipp-listener
file path=etc/apache2/2.2/httpd-standalone-ipp.conf \
file path=etc/apache/httpd-standalone-ipp.conf \
    original_name=SUNWippl:etc/apache/httpd-standalone-ipp.conf preserve=true
file path=lib/svc/manifest/application/print/ipp-listener.xml group=sys \
    mode=0444
file path=usr/apache2/2.2/libexec/mod_ipp.so mode=0555
file path=usr/apache/libexec/mod_ipp.so mode=0555
file path=var/lp/ipp-listener/index.html mode=0444
legacy pkg=SUNWipplr \
    desc="Internet Printing Protocol(IPP) Apache configuration for service modul
    name="Internet Printing Protocol(IPP) listener, (root)" \
```

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