

```

*****
54398 Tue Jan 28 08:41:44 2014
new/usr/src/cmd/lofiadm/main.c
3015 lofiadm should use libz.so.1, not libz.so
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22 * Copyright 2009 Sun Microsystems, Inc. All rights reserved.
23 * Use is subject to license terms.
24 * Copyright 2012 Joyent, Inc. All rights reserved.
25 *
26 * Copyright 2013 Nexenta Systems, Inc. All rights reserved.
27 * Copyright (c) 2014 Gary Mills
28 */
29
30 /*
31 * lofiadm - administer lofi(7d). Very simple, add and remove file<->device
32 * associations, and display status. All the ioctls are private between
33 * lofi and lofiadm, and so are very simple - device information is
34 * communicated via a minor number.
35 */
36
37 #include <sys/types.h>
38 #include <sys/param.h>
39 #include <sys/lofi.h>
40 #include <sys/stat.h>
41 #include <sys/sysmacros.h>
42 #include <netinet/in.h>
43 #include <stdio.h>
44 #include <fcntl.h>
45 #include <locale.h>
46 #include <string.h>
47 #include <strings.h>
48 #include <errno.h>
49 #include <stdlib.h>
50 #include <unistd.h>
51 #include <stropts.h>
52 #include <libdevinfo.h>
53 #include <libgen.h>
54 #include <ctype.h>
55 #include <dlfcn.h>
56 #include <limits.h>
57 #include <security/cryptoki.h>
58 #include <cryptoutil.h>
59 #include <sys/crypto/ioctl.h>
60 #include <sys/crypto/ioctladmin.h>
61 #include "utils.h"

```

```

62 #include <LzmaEnc.h>
63
64 /* Only need the IV len #defines out of these files, nothing else. */
65 #include <aes/aes_impl.h>
66 #include <des/des_impl.h>
67 #include <blowfish/blowfish_impl.h>
68
69 static const char USAGE[] =
70     "Usage: %s [-r] -a file [ device ]\n"
71     "         %s [-r] -c crypto_algorithm -a file [device]\n"
72     "         %s [-r] -c crypto_algorithm -k raw_key_file -a file [device]\n"
73     "         %s [-r] -c crypto_algorithm -T [token]:[manuf]:[serial]:key "
74     "         -a file [device]\n"
75     "         %s [-r] -c crypto_algorithm -T [token]:[manuf]:[serial]:key "
76     "         -k wrapped_key_file -a file [device]\n"
77     "         %s [-r] -c crypto_algorithm -e -a file [device]\n"
78     "         %s -d file | device\n"
79     "         %s -C [gzip|gzip-6|gzip-9|lzma] [-s segment_size] file\n"
80     "         %s -U file\n"
81     "         %s [ file | device ]\n";
82
83 typedef struct token_spec {
84     char *name;
85     char *mfr;
86     char *serno;
87     char *key;
88 } token_spec_t;
89
90 #define UNCHANGED_PORTION_OMITTED
91
92
93
94 /* For displaying lofi mappings */
95 #define FORMAT "%-20s %s %-30s %s\n"
96
97 #define COMPRESS_ALGORITHM "gzip"
98 #define COMPRESS_THRESHOLD 2048
99 #define SEGSIZE 131072
100 #define BLOCK_SIZE 512
101 #define KILOBYTE 1024
102 #define MEGABYTE (KILOBYTE * KILOBYTE)
103 #define GIGABYTE (KILOBYTE * MEGABYTE)
104 #define LIBZ "libz.so.1"
105 #define LIBZ "libz.so"
106
107 static void
108 usage(const char *pname)
109 {
110     (void) fprintf(stderr, gettext(USAGE), pname, pname, pname,
111                   pname, pname, pname, pname, pname, pname,
112                   pname, pname);
113     exit(E_USAGE);
114 }
115
116 static int
117 gzip_compress(void *src, size_t srclen, void *dst, size_t *dstlen, int level)
118 {
119     static int (*compress2p)(void *, ulong_t *, void *, size_t, int) = NULL;
120     void *libz_hdl = NULL;
121
122     /*
123      * The first time we are called, attempt to dlopen()
124      * libz.so.1 and get a pointer to the compress2() function
125      * libz.so and get a pointer to the compress2() function
126      */
127     if (compress2p == NULL) {
128         if ((libz_hdl = openlib(LIBZ)) == NULL)
129             die(gettext("could not find %s. "),
130               "gzip compression unavailable\n"), LIBZ);
131     }

```

```
179         if ((*compress2p =
180             (int (*)(void *, ulong_t *, void *, size_t, int))
181             dlsym(libz_hdl, "compress2")) == NULL) {
182             closelib();
183             die(gettext("could not find the correct %s. "
184                 "gzip compression unavailable\n"), LIBZ);
185         }
186     }

188     if ((*compress2p)(dst, (ulong_t *)dstlen, src, srclen, level) != 0)
189         return (-1);
190     return (0);
191 }
```

unchanged_portion_omitted