

new/usr/src/cmd/certsdir/Makefile

1

```
*****
      890 Fri Jun 15 16:18:43 2012
new/usr/src/cmd/certsdir/Makefile
add code to autopopulate OpenSSL hash link directory at boot
*****
```

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #
11
12 #
13 # Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
14 #
15 PROG=          certsdir
16
17 OBJS=          certsdir.o utils.o scf.o
18 SRCS=          $(OBJS:%.o=%.c)
19
20 include ../Makefile.cmd
21
22 FILEMODE =     0555
23
24 MANIFEST =     certsdir.xml
25 ROOTMANIFESTDIR = $(ROOTSVCSYSTEM)
26
27 LDLIBS +=      -lscf
28
29 .KEEP_STATE:
30
31 .PARALLEL: $(OBJS)
32
33 all:          $(PROG)
34
35 $(PROG):      $(OBJS)
36               $(LINK.c) $(OBJS) -o $$@ $(LDLIBS)
37               $(POST_PROCESS)
38
39 install: all .WAIT $(ROOTLIBPROG) $(ROOTMANIFEST)
40
41 lint: lint_SRCS
42
43 clean:
44         $(RM) $(OBJS)
45
46 include ../Makefile.targ
```

new/usr/src/cmd/certsdir/certsdir.c

1

```
*****
9428 Fri Jun 15 16:18:44 2012
new/usr/src/cmd/certsdir/certsdir.c
add code to autopopulate OpenSSL hash link directory at boot
*****
```

```
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21
22 /*
23  * Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
24  */
```

```
26 #include <stdlib.h>
27 #include <string.h>
28 #include <strings.h>
29 #include <errno.h>
30 #include <unistd.h>
31 #include <limits.h>
32 #include <fcntl.h>
33 #include <err.h>
34 #include <locale.h>
35 #include <libintl.h>
36 #include <sys/types.h>
37 #include <sys/stat.h>
38 #include <sys/wait.h>
39 #include <dirent.h>
40 #include <stddef.h>
41 #include <libscf.h>
```

```
43 #include "utils.h"
44 #include "scf.h"
```

```
46 #ifndef TEXT_DOMAIN
47 #define TEXT_DOMAIN "SYS_TEST"
48 #endif
```

```
50 #define _(x) gettext(x)
```

```
52 #define CERTS_FILE_DIR "/etc/certs/CA"
53 #define CERTS_LINK_DIR "/etc/openssl/certs"
```

```
55 #define OPENSOLARIS "/usr/bin/openssl"
```

```
57 #define FMT_VERSION "OpenSSL Version: %s (%s)\n"
58 #define MSG_OLD_HASH "supports old hash only"
59 #define MSG_NEW_HASH "supports new hashes"
```

```
61 #define FMT_WRONG "WRONG: [%s] -> %s\n should be: %s\n"
```

new/usr/src/cmd/certsdir/certsdir.c

2

```
62 #define FMT_MISSING "MISSING: [%s] -> %s\n"
63 #define FMT_DANGLING "DANGLING: [%s] -> %s\n"
64 #define FMT_UNKNOWN "UNKNOWN TARGET: [%s] -> %s\n"
```

```
66 #define MAX_HASHES 2
```

```
68 typedef struct cert_file {
69     char *path;
70     char *hash[MAX_HASHES];
71     uint8_t hashcnt;
72
73     struct cert_file *next;
74 } cert_file_t;
```

```
76 typedef struct cert_link {
77     char *path;
78     char *hash;
79     uint8_t targetfound;
80
81     struct cert_link *next;
82 } cert_link_t;
```

```
84 /* configuration from smf(5) */
85 static char *certs_file_dir = CERTS_FILE_DIR;
86 static char *certs_link_dir = CERTS_LINK_DIR;
87 static char *openssl = OPENSOLARIS;
88 static int verbose = B_FALSE;
89 static boolean_t remove_dangling = B_TRUE;
90 static boolean_t remove_unknown = B_FALSE;
91 static boolean_t create_missing = B_TRUE;
92 static boolean_t correct_wrong = B_TRUE;
```

```
94 /* list heads */
95 static cert_file_t *file0 = NULL;
96 static cert_link_t *link0 = NULL;
```

```
98 #define OP_HASH 0
99 #define OP_VERSION 1
100 #define OP_HASH_OLD 2
101 #define NUM_OPS 3
102 char *
103 run_openssl(unsigned int op, char *arg)
104 {
105     pid_t pid;
106     int des[2];
107     int i;
108     ssize_t num;
109     char buf[500];
110     int status;
```

```
112     if (op >= NUM_OPS)
113         abort();
114
115     (void) pipe(des);
116     switch (pid = fork()) {
117     case -1: /* error */
118         err(SMF_EXIT_ERR_FATAL, _("forking child for openssl"));
119         break;
120     case 0: /* child */
121         (void) close(des[0]);
122         if (dup2(des[1], STDOUT_FILENO) == -1 || dup2(des[1],
123             STDERR_FILENO) == -1)
124             err(100, NULL);
125         (void) close(STDIN_FILENO);
126         switch (op) {
127         case OP_HASH:
```

```

128         execlp(openssl, openssl, "x509", "-hash",
129               "-noout", "-in", arg, (char *)0);
130         break;
131     case OP_VERSION:
132         execlp(openssl, openssl, "version", (char *)0);
133         break;
134     case OP_HASH_OLD:
135         execlp(openssl, openssl, "x509",
136               "-subject_hash_old", "-noout", "-in",
137               arg, (char *)0);
138         break;
139     }
140     printf(_("exec openssl failed: %s\n"), strerror(errno));
141     exit(SMF_EXIT_ERR_FATAL);
142     break;
143 default: /* parent */
144     (void) close(des[1]);
145     num = read(des[0], &buf, sizeof (buf));
146     (void) close(des[0]);

148     if (num < 0)
149         err(101, _("reading from openssl child"));

151     buf[num] = 0;
152     for (i = 0; i < num; i++) {
153         if (buf[i] == '\n' || buf[i] == '\r') {
154             buf[i] = 0;
155             break;
156         }
157     }

159     if (waitpid(pid, &status, 0) == -1 || !WIFEXITED(status))
160         err(SMF_EXIT_ERR_FATAL,
161             _("waiting for openssl child"));
162     if (WEXITSTATUS(status) != 0)
163         errx(SMF_EXIT_ERR_FATAL,
164             _("error from openssl child (rc: %d): %s"),
165             WEXITSTATUS(status), buf);

167     return (xstrdup(buf));
168 }
169 }

171 int newossl = 0;

174 void
175 detect_openssl(void)
176 {
177     char *vers;

179     /* determine openssl version */
180     vers = run_openssl(OP_VERSION, NULL);
181     if (strstr(vers, "OpenSSL 1.") == vers)
182         newossl = 1;
183     else if (strstr(vers, "OpenSSL 0.9.") != vers)
184         errx(SMF_EXIT_ERR_FATAL, _("unrecognised OpenSSL: %s"), vers);
185     if (verbose)
186         printf(_(FMT_VERSION), vers, (newossl ? _(MSG_NEW_HASH) :
187             _(MSG_OLD_HASH)));
188     free(vers);
189 }

191 void
192 store_certlink(cert_link_t *cd)
193 {

```

```

194     if (link0 == NULL) {
195         link0 = cd;
196     } else {
197         cert_link_t *t = link0;
198         while (t->next != NULL)
199             t = t->next;
200         t->next = cd;
201     }
202 }

204 cert_link_t *
205 find_link_by_hash(char *hash)
206 {
207     cert_link_t *t = link0;
208     while (t != NULL) {
209         if (strcmp(t->hash, hash) == 0)
210             return (t);
211         t = t->next;
212     }
213     return (NULL);
214 }

216 void
217 store_certfile(cert_file_t *cs)
218 {
219     if (file0 == NULL) {
220         file0 = cs;
221     } else {
222         cert_file_t *t = file0;
223         while (t->next != NULL)
224             t = t->next;
225         t->next = cs;
226     }
227 }

229 char *
230 add_dot_zero(char *str)
231 {
232     char *ret;
233     xasprintf(&ret, "%s.0", str);
234     free(str);
235     return (ret);
236 }

238 void
239 populate_hash_list(cert_file_t *cs)
240 {
241     if (cs->path[0] != '/')
242         abort();

244     if (newossl)
245         cs->hash[cs->hashcnt++] =
246             add_dot_zero(run_openssl(OP_HASH_OLD, cs->path));
247     cs->hash[cs->hashcnt++] = add_dot_zero(run_openssl(OP_HASH, cs->path));
248 }

250 void
251 get_cert_file_list(void)
252 {
253     DIR *dir;
254     struct dirent *de;

256     dir = opendir(certs_file_dir);
257     if (dir == NULL)
258         err(SMF_EXIT_ERR_FATAL, _("could not read directory %s"),
259             certs_file_dir);

```

```

261     for (de = readdir(dir); de != NULL; de = readdir(dir)) {
262         char *a;
263         cert_file_t *cs;
264
265         if (strcmp(".", de->d_name) == 0 ||
266             strcmp("../", de->d_name) == 0)
267             continue;
268
269         cs = xmalloc(sizeof(*cs));
270         xasprintf(&a, "%s/%s", certs_file_dir, de->d_name);
271         cs->path = xrealpath(a);
272         free(a);
273         populate_hash_list(cs);
274         store_certfile(cs);
275     }
276     (void) closedir(dir);
277 }
278
279 void
280 removehashlink(char *hash)
281 {
282     char *fqhash;
283     xasprintf(&fqhash, "%s/%s", certs_link_dir, hash);
284     xunlink(fqhash);
285     free(fqhash);
286 }
287
288 void
289 fixhashlink(char *file, char *hash)
290 {
291     char *fqhash;
292     char *relfile;
293
294     xasprintf(&fqhash, "%s/%s", certs_link_dir, hash);
295     relfile = xmakerelative(certs_link_dir, file);
296
297     if (xexists(fqhash))
298         xunlink(fqhash);
299
300     if (symlink(relfile, fqhash) != 0)
301         err(SMF_EXIT_ERR_FATAL, _("could not create symlink %s"),
302            fqhash);
303
304     free(fqhash);
305     free(relfile);
306 }
307
308 void
309 get_cert_link_list(void)
310 {
311     DIR *dir;
312     struct dirent *de;
313
314     dir = opendir(certs_link_dir);
315     if (dir == NULL)
316         err(SMF_EXIT_ERR_FATAL, _("could not read directory %s"),
317            certs_link_dir);
318
319     for (de = readdir(dir); de != NULL; de = readdir(dir)) {
320         cert_link_t *cd;
321         char *a, *b;
322
323         if (strcmp(".", de->d_name) == 0 ||
324             strcmp("../", de->d_name) == 0)
325             continue;

```

```

327         cd = xmalloc(sizeof(*cd));
328
329         cd->hash = xstrdup(de->d_name);
330
331         xasprintf(&a, "%s/%s", certs_link_dir, de->d_name);
332         b = xreadlink(a);
333         free(a);
334         if (b[0] == '.') {
335             xasprintf(&a, "%s/%s", certs_link_dir, b);
336             free(b);
337             b = a;
338         }
339         a = xrealpath(b);
340         free(b);
341         cd->path = a;
342
343         store_certlink(cd);
344     }
345     (void) closedir(dir);
346 }
347
348 void
349 read_config(void)
350 {
351     int a;
352     char *b;
353
354     a = get_config_boolean("verbose");
355     if (a >= 0)
356         verbose = a;
357     a = get_config_boolean("remove_dangling");
358     if (a >= 0)
359         remove_dangling = a;
360     a = get_config_boolean("remove_unknown");
361     if (a >= 0)
362         remove_unknown = a;
363     a = get_config_boolean("create_missing");
364     if (a >= 0)
365         create_missing = a;
366     a = get_config_boolean("correct_wrong");
367     if (a >= 0)
368         correct_wrong = a;
369
370     b = get_config_string("certs_file_dir");
371     if (b != NULL)
372         certs_file_dir = b;
373     b = get_config_string("certs_link_dir");
374     if (b != NULL)
375         certs_link_dir = b;
376     b = get_config_string("openssl_command");
377     if (b != NULL)
378         openssl = b;
379 }
380
381 int
382 main(void)
383 {
384     cert_file_t *tf;
385     cert_link_t *tl;
386
387     (void) setlocale(LC_ALL, "");
388     (void) textdomain(TEXT_DOMAIN);
389
390     if (init_scf() != 0)

```

```

392         errx(SMF_EXIT_ERR_FATAL, _("could not connect to smf(5): %s"),
393             scf_strerror(scf_error()));
394     read_config();
395     fini_scf();

397 #ifdef DEBUG
398     verbose = B_TRUE;
399 #endif

401     detect_openssl();

403     get_cert_link_list();
404     get_cert_file_list();

406     /*
407      * look at each certificate file and determine if there is
408      * a symlink to it in the link directory
409      */
410     for (tf = file0; tf != NULL; tf = tf->next) {
411         int j;
412         for (j = 0; j < tf->hashcnt; j++) {
413             t1 = find_link_by_hash(tf->hash[j]);
414             if (t1) {
415                 t1->targetfound = 1;
416                 if (correct_wrong && strcmp(t1->path, tf->path)
417                     != 0) {
418                     if (verbose)
419                         printf(_(FMT_WRONG),
420                             tf->hash[j], t1->path,
421                             tf->path);
422                     fixhashlink(tf->path, tf->hash[j]);
423                 }
424             } else if (create_missing) {
425                 if (verbose)
426                     printf(_(FMT_MISSING), tf->hash[j],
427                         tf->path);
428                 fixhashlink(tf->path, tf->hash[j]);
429             }
430         }
431     }
432     /*
433      * look at each certificate link and determine if it's dangling or
434      * pointing to a certificate file which we haven't seen
435      */
436     for (t1 = link0; t1 != NULL; t1 = t1->next) {
437         if (!xexists(t1->path)) {
438             if (remove_dangling) {
439                 if (verbose)
440                     printf(_(FMT_DANGLING), t1->hash,
441                         t1->path);
442                 removehashlink(t1->hash);
443             }
444         } else if (!t1->targetfound) {
445             if (remove_unknown) {
446                 if (verbose)
447                     printf(_(FMT_UNKNOWN), t1->hash,
448                         t1->path);
449                 removehashlink(t1->hash);
450             }
451         }
452     }

454     return (0);
455 }

```

new/usr/src/cmd/certsdir/certsdir.xml

1

```
*****
3371 Fri Jun 15 16:18:44 2012
new/usr/src/cmd/certsdir/certsdir.xml
add code to autopopulate OpenSSL hash link directory at boot
*****

1 <?xml version="1.0"?>
2 <!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
3 <!--
4 Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>

6 CDDL HEADER START

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

12 You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
13 or http://www.opensolaris.org/os/licensing.
14 See the License for the specific language governing permissions
15 and limitations under the License.

17 When distributing Covered Code, include this CDDL HEADER in each
18 file and include the License file at usr/src/OPENSOLARIS.LICENSE.
19 If applicable, add the following below this CDDL HEADER, with the
20 fields enclosed by brackets "[]" replaced with your own identifying
21 information: Portions Copyright [yyyy] [name of copyright owner]

23 CDDL HEADER END

25 NOTE: This service manifest is not editable; its contents will
26 be overwritten by package or patch operations, including
27 operating system upgrade. Make customizations in a different
28 file.
29 -->

31 <service_bundle type='manifest' name='crypto-ca-certificates:certsdir'>

33 <service
34   name='system/certsdir'
35   type='service'
36   version='1'>

38   <create_default_instance enabled='true' />

40   <single_instance />

42   <dependency
43     name='filesystem-minimal'
44     type='service'
45     grouping='require_all'
46     restart_on='none'>
47     <service_fmri value='svc:/system/filesystem/minimal' />
48   </dependency>

50   <exec_method
51     type='method'
52     name='start'
53     exec='/usr/lib/certsdir'
54     timeout_seconds='60' />

56   <exec_method
57     type='method'
58     name='stop'
59     exec=':true'
60     timeout_seconds='0' />
```

new/usr/src/cmd/certsdir/certsdir.xml

2

```
62   <exec_method
63     type='method'
64     name='refresh'
65     exec='/usr/lib/certsdir'
66     timeout_seconds='60' />

68   <property_group name='startd' type='framework'>
69     <propval name='duration' type='astring' value='transient' />
70   </property_group>

72   <property_group name='config' type='application'>
73     <propval name='verbose' type='boolean' value='false' />
74     <!-- remove symlinks from the link directory that
75          point to non-existent files: -->
76     <propval name='remove_dangling' type='boolean' value='true' />
77     <!-- remove symlinks from the link directory that
78          point to files from somewhere outside the CA file directory
79     <propval name='remove_unknown' type='boolean' value='false' />
80     <!-- create symlinks to files from the CA file directory that
81          are missing from the link directory -->
82     <propval name='create_missing' type='boolean' value='true' />
83     <!-- recreate symlinks for files from the CA file directory that
84          are linked with an incorrect hash in the link directory -->
85     <propval name='correct_wrong' type='boolean' value='true' />
86     <!-- the directory with CA certificate files -->
87     <propval name='certs_file_dir' type='astring' value='/etc/certs/'
88     <!-- the directory with OpenSSL certificate hash symlinks -->
89     <propval name='certs_link_dir' type='astring' value='/etc/openssl'
90     <!-- the OpenSSL binary to use -->
91     <propval name='openssl' type='astring' value='/usr/bin/openssl'
92   </property_group>

94   <stability value='Unstable' />

96   <template>
97     <common_name>
98       <loctext xml:lang='C'>
99   certsdir: OpenSSL certificates directory symlink populator
100     </loctext>
101   </common_name>
102   </template>
103 </service>

105 </service_bundle>
```

new/usr/src/cmd/certsdir/scf.c

1

```
*****
4446 Fri Jun 15 16:18:44 2012
new/usr/src/cmd/certsdir/scf.c
add code to autopopulate OpenSSL hash link directory at boot
*****

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

22 /*
23  * Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
24  */

26 #include <stdio.h>
27 #include <stdlib.h>
28 #include <unistd.h>
29 #include <string.h>
30 #include <err.h>
31 #include <limits.h>
32 #include <libscf.h>

34 #include "utils.h"
35 #include "scf.h"

37 static char *fmri;
38 static scf_handle_t *scf;
39 static scf_instance_t *inst;
40 static scf_propertygroup_t *pg;

42 /*
43  * Initialise our connection to smf(5). Returns:
44  * 0 on success
45  * <0 for any failure
46  * -2 if we believe we're not actually running under smf(5)
47  */
48 int
49 init_scf(void)
50 {
51     int rc = -1;
52     int fmrisz;

54     scf = scf_handle_create(SCF_VERSION);
55     if (scf == NULL)
56         return (-1);
57     if (scf_handle_bind(scf) != 0)
58         goto cleanup0;

60     if (getenv("SMF_TEST_FMRI") != NULL) {
61         fprintf(stderr, "DEBUG: smf test mode engaged\n");
```

new/usr/src/cmd/certsdir/scf.c

2

```
62         fmri = strdup(getenv("SMF_TEST_FMRI"));
63         goto have_fmri;
64     }

66     fmrisz = scf_myname(scf, NULL, 0);
67     if (fmrisz == -1) {
68         if (scf_error() == SCF_ERROR_NOT_SET) {
69             rc = -2;
70         }
71         goto cleanup1;
72     }
73     fmri = xmalloc(fmrisz + 1);
74     fmrisz = scf_myname(scf, fmri, fmrisz + 1);
75     if (fmrisz == -1)
76         goto cleanup2;

78 have_fmri:

80     if ((inst = scf_instance_create(scf)) == NULL ||
81         (pg = scf_pg_create(scf)) == NULL)
82         goto cleanup3;

84     if (scf_handle_decode_fmri(scf, fmri, NULL, NULL, inst,
85         NULL, NULL, SCF_DECODE_FMRI_EXACT) != 0)
86         goto cleanup3;

88     if (scf_instance_get_pg_composed(inst, NULL, "config", pg) != 0)
89         goto cleanup3;

91     return (0);

93 cleanup3:
94     if (pg != NULL) {
95         scf_pg_destroy(pg);
96         pg = NULL;
97     }
98     if (inst != NULL) {
99         scf_instance_destroy(inst);
100         inst = NULL;
101     }
102 cleanup2:
103     free(fmri);
104     fmri = NULL;
105 cleanup1:
106     (void) scf_handle_unbind(scf);
107 cleanup0:
108     (void) scf_handle_destroy(scf);
109     scf = NULL;
110     return (rc);
111 }

113 void
114 fini_scf(void)
115 {
116     (void) scf_pg_destroy(pg);
117     pg = NULL;
118     (void) scf_instance_destroy(inst);
119     inst = NULL;
120     free(fmri);
121     fmri = NULL;
122     (void) scf_handle_unbind(scf);
123     (void) scf_handle_destroy(scf);
124     scf = NULL;
125 }

127 /*
```

```

128 * Fetch the boolean value of a property from the 'config' property
129 * group in our smf(5) instance.
130 * Returns:
131 *   <0 on failure
132 *   0 if found and false
133 *   1 if found and true
134 */
135 int
136 get_config_boolean(char *name)
137 {
138     scf_property_t *prop = NULL;
139     scf_value_t *val = NULL;
140     uint8_t out;
141     int rc = -1;

143     prop = scf_property_create(scf);
144     val = scf_value_create(scf);
145     if (prop == NULL || val == NULL)
146         goto cleanup0;

148     if (scf_pg_get_property(pg, name, prop) != 0)
149         goto cleanup0;

151     if (scf_property_is_type(prop, SCF_TYPE_BOOLEAN) != 0)
152         goto cleanup0;

154     if (scf_property_get_value(prop, val) != 0)
155         goto cleanup0;

157     if (scf_value_get_boolean(val, &out) != 0)
158         goto cleanup0;

160     rc = out;

162 cleanup0:
163     if (val != NULL)
164         scf_value_destroy(val);
165     if (prop != NULL)
166         scf_property_destroy(prop);

168     return (rc);
169 }

171 /*
172 * Fetch the boolean value of a property from the 'config' property
173 * group in our smf(5) instance.
174 * Returns:
175 *   NULL on failure
176 *   a NUL-terminated string value on success, caller to free
177 */
178 char *
179 get_config_string(char *name)
180 {
181     scf_property_t *prop = NULL;
182     scf_value_t *val = NULL;
183     char out[PATH_MAX];
184     char *rc = NULL;

186     prop = scf_property_create(scf);
187     val = scf_value_create(scf);
188     if (prop == NULL || val == NULL)
189         goto cleanup0;

191     if (scf_pg_get_property(pg, name, prop) != 0)
192         goto cleanup0;

```

```

194     if (scf_property_is_type(prop, SCF_TYPE_ASTRING) != 0)
195         goto cleanup0;

197     if (scf_property_get_value(prop, val) != 0)
198         goto cleanup0;

200     if (scf_value_get_astring(val, out, sizeof (out)) == -1)
201         goto cleanup0;

203     rc = xstrdup(out);

205 cleanup0:
206     if (val != NULL)
207         scf_value_destroy(val);
208     if (prop != NULL)
209         scf_property_destroy(prop);

211     return (rc);
212 }

```


new/usr/src/cmd/certsdir/scf.h

1

```
*****
1109 Fri Jun 15 16:18:44 2012
new/usr/src/cmd/certsdir/scf.h
add code to autopopulate OpenSSL hash link directory at boot
*****
```

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

22 /*
23  * Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
24  */

26 #ifndef _SCF_H
27 #define _SCF_H

29 #ifdef __cplusplus
30 extern "C" {
31 #endif

33 int init_scf(void);
34 void fini_scf(void);
35 int get_config_boolean(char *);
36 char *get_config_string(char *);

38 #ifdef __cplusplus
39 }
40 #endif

42 #endif /* _SCF_H */
```

new/usr/src/cmd/certsdir/utills.c

1

```
*****
3807 Fri Jun 15 16:18:44 2012
new/usr/src/cmd/certsdir/utills.c
add code to autopopulate OpenSSL hash link directory at boot
*****
```

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

22 /*
23  * Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
24  */
```

```
26 #include <stdlib.h>
27 #include <string.h>
28 #include <strings.h>
29 #include <errno.h>
30 #include <unistd.h>
31 #include <limits.h>
32 #include <fcntl.h>
33 #include <err.h>
34 #include <libintl.h>
35 #include <sys/types.h>
36 #include <sys/stat.h>
37 #include <sys/wait.h>
38 #include <dirent.h>
39 #include <stddef.h>
40 #include <libscf.h>
```

```
42 #include "utills.h"
```

```
44 #define _(x)    gettext(x)
```

```
46 void *
47 xmalloc(size_t sz)
48 {
49     void *x = malloc(sz);
50     if (x == NULL)
51         abort();
52     bzero(x, sz);
53     return (x);
54 }
```

```
56 char *
57 xstrdup(const char *a)
58 {
59     char *x = strdup(a);
60     if (x == NULL)
61         abort();
```

new/usr/src/cmd/certsdir/utills.c

2

```
62     return (x);
63 }

65 int
66 xasprintf(char **str, const char *format, ...)
67 {
68     va_list ap;
69     int ret;

71     *str = NULL;
72     va_start(ap, format);
73     ret = vasprintf(str, format, ap);
74     va_end(ap);

76     if (ret == -1)
77         abort();

79     return (ret);
80 }

82 int
83 xexists(char *path)
84 {
85     struct stat buf;
86     if (stat(path, &buf) != 0) {
87         if (errno == ENOENT)
88             return (0);
89         else
90             err(SMF_EXIT_ERR_FATAL, _("stat on path: %s"), path);
91     }
92     return (1);
93 }

95 char *
96 xreadlink(char *path)
97 {
98     char tmp[PATH_MAX];
99     int len;

101     len = readlink(path, tmp, sizeof (tmp) - 1);
102     if (len == -1 || len >= (int)sizeof (tmp))
103         err(SMF_EXIT_ERR_FATAL, _("readlink failure on path: %s"),
104             path);
105     tmp[len] = '\0'; /* NUL-terminate what readlink gives us */

107     return (xstrdup(tmp));
108 }

110 char *
111 xrealpath(char *path)
112 {
113     char *ret;
114     char tmp[PATH_MAX];

116     ret = realpath(path, tmp);
117     if (ret == NULL) {
118         if (errno == ENOENT)
119             return (xstrdup(path));
120         else
121             err(SMF_EXIT_ERR_FATAL, _("realpath failure on"
122                 " path: %s"), path);
123     }

125     return (xstrdup(tmp));
126 }
```

```
128 void
129 xunlink(const char *path)
130 {
131     if (unlink(path) != 0)
132         err(SMF_EXIT_ERR_FATAL, _("could not unlink: %s"), path);
133 }

135 char *
136 xmakerelative(char *relativetodir, char *path)
137 {
138     int i;
139     int comlen = 0;
140     int extraelems = 0;
141     char tmp[PATH_MAX];
142     char *lastsrc, *lastpath, *srccomp, *pathcomp;
143     char *trelativetodir = strdup(relativetodir);
144     char *tpath = strdup(path);

146     if (relativetodir[0] != '/' || path[0] != '/')
147         abort();

149     srccomp = strtok_r(trelativetodir, "/", &lastsrc);
150     pathcomp = strtok_r(tpath, "/", &lastpath);

152     /* discard elements common to both paths */
153     while (srccomp != NULL && pathcomp != NULL &&
154           strcmp(srccomp, pathcomp) == 0) {
155         srccomp = strtok_r(NULL, "/", &lastsrc);
156         pathcomp = strtok_r(NULL, "/", &lastpath);
157     }
158     /* record the start of the unique part of the target path */
159     comlen = pathcomp - tpath;
160     /* count the number of non-common path elements in the target dir */
161     while (srccomp != NULL) {
162         extraelems++;
163         srccomp = strtok_r(NULL, "/", &lastsrc);
164     }

166     if (extraelems * 3 + strlen(&path[comlen]) + 1 > PATH_MAX)
167         errx(17, "relative path generation error");

169     /*
170      * generate a path that escapes only the non-common elements of the
171      * target directory
172      */
173     tmp[0] = '\0';
174     for (i = 0; i < extraelems; i++)
175         strcat(tmp, "../");
176     strcat(tmp, &path[comlen]);

178     free(trelativetodir);
179     free(tpath);

181     return (strdup(tmp));
182 }
```

new/usr/src/cmd/certsdir/utills.h

1

```
*****
1240 Fri Jun 15 16:18:44 2012
new/usr/src/cmd/certsdir/utills.h
add code to autopopulate OpenSSL hash link directory at boot
*****
```

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

22 /*
23  * Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
24  */

26 #ifndef _UTILS_H
27 #define _UTILS_H

29 #ifdef __cplusplus
30 extern "C" {
31 #endif

33 void *xmalloc(size_t);
34 char *xstrdup(const char *);
35 int xasprintf(char **, const char *, ...);
36 int xexists(char *);
37 char *xreadlink(char *);
38 char *xrealpath(char *);
39 void xunlink(const char *);
40 char *xmakerelative(char *, char *);

42 #ifdef __cplusplus
43 }
44 #endif

46 #endif /* _UTILS_H */
```

new/usr/src/cmd/cmd-crypto/etc/Makefile.ca-links

1

6658 Fri Jun 15 16:18:45 2012

new/usr/src/cmd/cmd-crypto/etc/Makefile.ca-links

dont generate links in proto

```
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright (c) 2012, OmniTI Computer Consulting, Inc. All rights reserved.
24 #
25 #
26 # These CA certs are extracted from the NSS database libnssckbi.so.
27 # Each CA cert is pointed by a symbolic link. The name of the
28 # symbolic link file is determined by the "hash" value using
29 # openssl command. When the NSS database is updated, we will extract
30 # the CA certs out of it and deliver the new set of CA certs.
31 #
32 #
33 CASRCDIR =      CA-certs
34 OPENSSL =      /usr/bin/openssl
35 #
36 CAFILES = ABAecom_sub._Am._Bankers_Assn._Root_CA.pem \
37 AddTrust_External_Root.pem \
38 AddTrust_Low-Value_Services_Root.pem \
39 AddTrust_Public_Services_Root.pem \
40 AddTrust_Qualified_Certificates_Root.pem \
41 America_Online_Root_Certification_Authority_1.pem \
42 America_Online_Root_Certification_Authority_2.pem \
43 AOL_Time_Warner_Root_Certification_Authority_1.pem \
44 AOL_Time_Warner_Root_Certification_Authority_2.pem \
45 Baltimore_CyberTrust_Root.pem \
46 beTRUSTed_Root_CA-Baltimore_Implementation.pem \
47 beTRUSTed_Root_CA_-_Entrust_Implementation.pem \
48 beTRUSTed_Root_CA.pem \
49 beTRUSTed_Root_CA_-_RSA_Implementation.pem \
50 Camerfirma_Chambers_of_Commerce_Root.pem \
51 Camerfirma_Global_Chambersign_Root.pem \
52 Certigna.pem \
53 Certplus_Class_2_Primary_CA.pem \
54 Certum_Root_CA.pem \
55 Comodo_AAA_Services_root.pem \
56 COMODO_Certification_Authority.pem \
57 COMODO_ECC_Certification_Authority.pem \
58 Comodo_Secure_Services_root.pem \
59 Comodo_Trusted_Services_root.pem \
60 ComSign_CA.pem \
61 ComSign_Secured_CA.pem \
```

new/usr/src/cmd/cmd-crypto/etc/Makefile.ca-links

2

```
62 Cybertrust_Global_Root.pem \
63 Deutsche_Telekom_Root_CA_2.pem \
64 DigiCert_Assured_ID_Root_CA.pem \
65 DigiCert_Global_Root_CA.pem \
66 DigiCert_High_Assurance_EV_Root_CA.pem \
67 Digital_Signature_Trust_Co._Global_CA_1.pem \
68 Digital_Signature_Trust_Co._Global_CA_2.pem \
69 Digital_Signature_Trust_Co._Global_CA_3.pem \
70 Digital_Signature_Trust_Co._Global_CA_4.pem \
71 DST_ACES_CA_X6.pem \
72 DST_Root_CA_X3.pem \
73 Entrust.net_Global_Secure_Personal_CA.pem \
74 Entrust.net_Global_Secure_Server_CA.pem \
75 Entrust.net_Premium_2048_Secure_Server_CA.pem \
76 Entrust.net_Secure_Personal_CA.pem \
77 Entrust.net_Secure_Server_CA.pem \
78 Entrust_Root_Certification_Authority.pem \
79 ePKI_Root_Certification_Authority.pem \
80 Equifax_Secure_CA.pem \
81 Equifax_Secure_eBusiness_CA_1.pem \
82 Equifax_Secure_eBusiness_CA_2.pem \
83 Equifax_Secure_Global_eBusiness_CA.pem \
84 Firmaprofesional_Root_CA.pem \
85 GeoTrust_Global_CA_2.pem \
86 GeoTrust_Global_CA.pem \
87 GeoTrust_Primary_Certification_Authority.pem \
88 GeoTrust_Universal_CA_2.pem \
89 GeoTrust_Universal_CA.pem \
90 GlobalSign_Root_CA.pem \
91 GlobalSign_Root_CA_-_R2.pem \
92 Go_Daddy_Class_2_CA.pem \
93 GTE_CyberTrust_Global_Root.pem \
94 GTE_CyberTrust_Root_CA.pem \
95 IGC_A.pem \
96 IPS_Chained_CAs_root.pem \
97 IPS_CLASE1_root.pem \
98 IPS_CLASE3_root.pem \
99 IPS_CLASEA1_root.pem \
100 IPS_CLASEA3_root.pem \
101 IPS_Servidores_root.pem \
102 IPS_Timestamping_root.pem \
103 MD5_Collisions_Forged_Rogue_CA_25c3.pem \
104 Microsec_e-Ssigno_Root_CA.pem \
105 NetLock_Business_Class_B_Root.pem \
106 NetLock_Express_Class_C_Root.pem \
107 NetLock_Notary_Class_A_Root.pem \
108 NetLock_Qualified_Class_QA_Root.pem \
109 Network_Solutions_Certificate_Authority.pem \
110 OISTE_WISEKey_Global_Root_GA_CA.pem \
111 QuoVadis_Root_CA_2.pem \
112 QuoVadis_Root_CA_3.pem \
113 QuoVadis_Root_CA.pem \
114 RSA_Root_Certificate_1.pem \
115 RSA_Security_1024_v3.pem \
116 RSA_Security_2048_v3.pem \
117 Secure_Global_CA.pem \
118 SecureTrust_CA.pem \
119 Security_Communication_EV_RootCA1.pem \
120 Security_Communication_Root_CA.pem \
121 Sonera_Class_1_Root_CA.pem \
122 Sonera_Class_2_Root_CA.pem \
123 Staat_der_Nederlanden_Root_CA.pem \
124 Starfield_Class_2_CA.pem \
125 StartCom_Certification_Authority.pem \
126 StartCom_Ltd..pem \
127 S-TRUST_Authentication_and_Encryption_Root_CA_2005_PN.pem \
```

```

128     Swisscom_Root_CA_1.pem \
129     SwissSign_Gold_CA_-_G2.pem \
130     SwissSign_Platinum_CA_-_G2.pem \
131     SwissSign_Silver_CA_-_G2.pem \
132     Taiwan_GRCA.pem \
133     TC_TrustCenter_Class_2_CA_II.pem \
134     TC_TrustCenter_Class_3_CA_II.pem \
135     TC_TrustCenter,_Germany,_Class_2_CA.pem \
136     TC_TrustCenter,_Germany,_Class_3_CA.pem \
137     TC_TrustCenter_Universal_CA_I.pem \
138     TDC_Internet_Root_CA.pem \
139     TDC_OCES_Root_CA.pem \
140     Thawte_Personal_Basic_CA.pem \
141     Thawte_Personal_Freemail_CA.pem \
142     Thawte_Personal_Premium_CA.pem \
143     Thawte_Premium_Server_CA.pem \
144     thawte_Primary_Root_CA.pem \
145     Thawte_Server_CA.pem \
146     Thawte_Time_Stamping_CA.pem \
147     TURKTRUST_Certificate_Services_Provider_Root_1.pem \
148     TURKTRUST_Certificate_Services_Provider_Root_2.pem \
149     UTN_DATACorp_SGC_Root_CA.pem \
150     UTN_USERFirst_Email_Root_CA.pem \
151     UTN_USERFirst_Hardware_Root_CA.pem \
152     UTN-USER_First-Network_Applications.pem \
153     UTN_USERFirst_Object_Root_CA.pem \
154     ValiCert_Class_1_VA.pem \
155     ValiCert_Class_2_VA.pem \
156     Verisign_Class_1_Public_Primary_Certification_Authority_-_G2.pem
157     Verisign_Class_1_Public_Primary_Certification_Authority_-_G3.pem
158     Verisign_Class_1_Public_Primary_Certification_Authority.pem \
159     Verisign_Class_2_Public_Primary_Certification_Authority_-_G2.pem
160     Verisign_Class_2_Public_Primary_Certification_Authority_-_G3.pem
161     Verisign_Class_2_Public_Primary_Certification_Authority.pem \
162     Verisign_Class_3_Public_Primary_Certification_Authority_-_G2.pem
163     Verisign_Class_3_Public_Primary_Certification_Authority_-_G3.pem
164     Verisign_Class_3_Public_Primary_Certification_Authority_-_G5.pem
165     Verisign_Class_3_Public_Primary_Certification_Authority.pem \
166     Verisign_Class_4_Public_Primary_Certification_Authority_-_G2.pem
167     Verisign_Class_4_Public_Primary_Certification_Authority_-_G3.pem
168     Verisign_RSA_Secure_Server_CA.pem \
169     Verisign_Time_Stamping_Authority_CA.pem \
170     Visa_eCommerce_Root.pem \
171     Visa_International_Global_Root_2.pem \
172     Wells_Fargo_Root_CA.pem \
173     WellsSecure_Public_Root_Certificate_Authority.pem \
174     XRamp_Global_CA_Root.pem

176 IETCCAFILES += $(CAFILES:%=$(ROOTETCCADIR)/%)

178 $(ROOTETCCADIR)/%: $(CASRCDIR)/%
179     $(INS.file)
179     $(INS.file); \
180     ROOTCALINK=$(ROOTETCCALINKDIR)/'$(OPENSSL) x509 -noout -hash -in $<'.0;
181     $(RM) $$ROOTCALINK; \
182     $(LN) -s $(CATARGDIR)/$(@F) $$ROOTCALINK
183     -ROOTCALINK=$(ROOTETCCALINKDIR)/'$(OPENSSL) x509 -noout -subject_hash_o1
184     test "$$ROOTCALINK" = "$(ROOTETCCALINKDIR)/.0" || $(RM) $$ROOTCALINK; \
185     test "$$ROOTCALINK" = "$(ROOTETCCALINKDIR)/.0" || $(LN) -s $(CATARGDIR)/

```

new/usr/src/pkg/manifests/crypto-ca-certificates.mf

1

```
*****
11488 Fri Jun 15 16:18:45 2012
new/usr/src/pkg/manifests/crypto-ca-certificates.mf
add code to autopopulate OpenSSL hash link directory at boot
*****

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

22 #
23 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2012, OmniTI Computer Consulting, Inc. All rights reserved.
25 # Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
26 #

28 set name=pkg.fmri value=pkg:/crypto/ca-certificates@$(PKGVERS)
29 set name=pkg.description \
30   value="System X.509 Certificate Trust Anchors for Public Certificate Authorit
31 set name=pkg.summary \
32   value="System X.509 Certificate Trust Anchors for Public Certificate Authorit
33 set name=info.classification \
34   value=org.opensolaris.category.2008:System/Security
35 set name=variant.arch value=$(ARCH)
36 dir path=etc group=sys
37 dir path=etc/certs group=sys
38 dir path=etc/certs/CA group=sys
39 dir path=etc/openssl group=sys
40 dir path=etc/openssl/certs group=sys
41 dir path=lib
42 dir path=lib/svc
43 dir path=lib/svc/manifest group=sys
44 dir path=lib/svc/manifest/system group=sys
45 dir path=usr group=sys
46 dir path=usr/lib
47 file path=etc/certs/CA/ABACOM_sub.,_Am._Bankers_Assn._Root_CA.pem group=sys
48 file path=etc/certs/CA/AOL_Time_Warner_Root_Certification_Authority_1.pem \
49   group=sys
50 file path=etc/certs/CA/AOL_Time_Warner_Root_Certification_Authority_2.pem \
51   group=sys
52 file path=etc/certs/CA/AddTrust_External_Root.pem group=sys
53 file path=etc/certs/CA/AddTrust_Low-Value_Services_Root.pem group=sys
54 file path=etc/certs/CA/AddTrust_Public_Services_Root.pem group=sys
55 file path=etc/certs/CA/AddTrust_Qualified_Certificates_Root.pem group=sys
56 file path=etc/certs/CA/America_Online_Root_Certification_Authority_1.pem \
57   group=sys
58 file path=etc/certs/CA/America_Online_Root_Certification_Authority_2.pem \
59   group=sys
60 file path=etc/certs/CA/Baltimore_CyberTrust_Root.pem group=sys
61 file path=etc/certs/CA/COMODO_Certification_Authority.pem group=sys
```

new/usr/src/pkg/manifests/crypto-ca-certificates.mf

2

```
62 file path=etc/certs/CA/COMODO_ECC_Certification_Authority.pem group=sys
63 file path=etc/certs/CA/Camerfirma_Chambbers_of_Commerce_Root.pem group=sys
64 file path=etc/certs/CA/Camerfirma_Global_Chambbersign_Root.pem group=sys
65 file path=etc/certs/CA/Certigna.pem group=sys
66 file path=etc/certs/CA/Certplus_Class_2_Primary_CA.pem group=sys
67 file path=etc/certs/CA/Certum_Root_CA.pem group=sys
68 file path=etc/certs/CA/ComSign_CA.pem group=sys
69 file path=etc/certs/CA/ComSign_Secured_CA.pem group=sys
70 file path=etc/certs/CA/Comodo_AAA_Services_root.pem group=sys
71 file path=etc/certs/CA/Comodo_Secure_Services_root.pem group=sys
72 file path=etc/certs/CA/Comodo_Trusted_Services_root.pem group=sys
73 file path=etc/certs/CA/Cybertrust_Global_Root.pem group=sys
74 file path=etc/certs/CA/DST_ACES_CA_X6.pem group=sys
75 file path=etc/certs/CA/DST_Root_CA_X3.pem group=sys
76 file path=etc/certs/CA/Deutsche_Telekom_Root_CA_2.pem group=sys
77 file path=etc/certs/CA/DigiCert_Assured_ID_Root_CA.pem group=sys
78 file path=etc/certs/CA/DigiCert_Global_Root_CA.pem group=sys
79 file path=etc/certs/CA/DigiCert_High_Assurance_EV_Root_CA.pem group=sys
80 file path=etc/certs/CA/Digital_Signature_Trust_Co._Global_CA_1.pem group=sys
81 file path=etc/certs/CA/Digital_Signature_Trust_Co._Global_CA_2.pem group=sys
82 file path=etc/certs/CA/Digital_Signature_Trust_Co._Global_CA_3.pem group=sys
83 file path=etc/certs/CA/Digital_Signature_Trust_Co._Global_CA_4.pem group=sys
84 file path=etc/certs/CA/Entrust.net_Global_Secure_Personal_CA.pem group=sys
85 file path=etc/certs/CA/Entrust.net_Global_Secure_Server_CA.pem group=sys
86 file path=etc/certs/CA/Entrust.net_Premium_2048_Secure_Server_CA.pem group=sys
87 file path=etc/certs/CA/Entrust.net_Secure_Personal_CA.pem group=sys
88 file path=etc/certs/CA/Entrust.net_Secure_Server_CA.pem group=sys
89 file path=etc/certs/CA/Entrust_Root_Certification_Authority.pem group=sys
90 file path=etc/certs/CA/Equifax_Secure_CA.pem group=sys
91 file path=etc/certs/CA/Equifax_Secure_Global_eBusiness_CA.pem group=sys
92 file path=etc/certs/CA/Equifax_Secure_eBusiness_CA_1.pem group=sys
93 file path=etc/certs/CA/Equifax_Secure_eBusiness_CA_2.pem group=sys
94 file path=etc/certs/CA/Firmaprofesional_Root_CA.pem group=sys
95 file path=etc/certs/CA/GTE_CyberTrust_Global_Root.pem group=sys
96 file path=etc/certs/CA/GTE_CyberTrust_Root_CA.pem group=sys
97 file path=etc/certs/CA/GeoTrust_Global_CA.pem group=sys
98 file path=etc/certs/CA/GeoTrust_Global_CA_2.pem group=sys
99 file path=etc/certs/CA/GeoTrust_Primary_Certification_Authority.pem group=sys
100 file path=etc/certs/CA/GeoTrust_Universal_CA.pem group=sys
101 file path=etc/certs/CA/GeoTrust_Universal_CA_2.pem group=sys
102 file path=etc/certs/CA/GlobalSign_Root_CA.pem group=sys
103 file path=etc/certs/CA/GlobalSign_Root_CA_-_R2.pem group=sys
104 file path=etc/certs/CA/Go_Daddy_Class_2_CA.pem group=sys
105 file path=etc/certs/CA/IGC_A.pem group=sys
106 file path=etc/certs/CA/IPS_CLASE1_root.pem group=sys
107 file path=etc/certs/CA/IPS_CLASE3_root.pem group=sys
108 file path=etc/certs/CA/IPS_CLASEA1_root.pem group=sys
109 file path=etc/certs/CA/IPS_CLASEA3_root.pem group=sys
110 file path=etc/certs/CA/IPS_Chained_CAs_root.pem group=sys
111 file path=etc/certs/CA/IPS_Servidores_root.pem group=sys
112 file path=etc/certs/CA/IPS_Timestamping_root.pem group=sys
113 file path=etc/certs/CA/MD5_Collisions_Forged_Rogue_CA_25c3.pem group=sys
114 file path=etc/certs/CA/Microsec_e-Szigno_Root_CA.pem group=sys
115 file path=etc/certs/CA/NetLock_Business_Class_B_Root.pem group=sys
116 file path=etc/certs/CA/NetLock_Express_Class_C_Root.pem group=sys
117 file path=etc/certs/CA/NetLock_Notary_Class_A_Root.pem group=sys
118 file path=etc/certs/CA/NetLock_Qualified_Class_QA_Root.pem group=sys
119 file path=etc/certs/CA/Network_Solutions_Certificate_Authority.pem group=sys
120 file path=etc/certs/CA/OISTE_WISEKey_Global_Root_GA_CA.pem group=sys
121 file path=etc/certs/CA/QuoVadis_Root_CA.pem group=sys
122 file path=etc/certs/CA/QuoVadis_Root_CA_2.pem group=sys
123 file path=etc/certs/CA/QuoVadis_Root_CA_3.pem group=sys
124 file path=etc/certs/CA/RSA_Root_Certificate_1.pem group=sys
125 file path=etc/certs/CA/RSA_Security_1024_v3.pem group=sys
126 file path=etc/certs/CA/RSA_Security_2048_v3.pem group=sys
127 file \
```

```
128 path=etc/certs/CA/S-TRUST_Authentication_and_Encryption_Root_CA_2005_PN.pem
129 group=sys
130 file path=etc/certs/CA/SecureTrust_CA.pem group=sys
131 file path=etc/certs/CA/Secure_Global_CA.pem group=sys
132 file path=etc/certs/CA/Security_Communication_EV_Root_CA1.pem group=sys
133 file path=etc/certs/CA/Security_Communication_Root_CA.pem group=sys
134 file path=etc/certs/CA/Sonera_Class_1_Root_CA.pem group=sys
135 file path=etc/certs/CA/Sonera_Class_2_Root_CA.pem group=sys
136 file path=etc/certs/CA/Staat_der_Nederlanden_Root_CA.pem group=sys
137 file path=etc/certs/CA/Starfield_Class_2_CA.pem group=sys
138 file path=etc/certs/CA/StartCom_Certification_Authority.pem group=sys
139 file path=etc/certs/CA/StartCom_Ltd..pem group=sys
140 file path=etc/certs/CA/SwissSign_Gold_CA_-_G2.pem group=sys
141 file path=etc/certs/CA/SwissSign_Platinum_CA_-_G2.pem group=sys
142 file path=etc/certs/CA/SwissSign_Silver_CA_-_G2.pem group=sys
143 file path=etc/certs/CA/Swisscom_Root_CA_1.pem group=sys
144 file path=etc/certs/CA/TC_TrustCenter,_Germany,_Class_2_CA.pem group=sys
145 file path=etc/certs/CA/TC_TrustCenter,_Germany,_Class_3_CA.pem group=sys
146 file path=etc/certs/CA/TC_TrustCenter_Class_2_CA_II.pem group=sys
147 file path=etc/certs/CA/TC_TrustCenter_Class_3_CA_II.pem group=sys
148 file path=etc/certs/CA/TC_TrustCenter_Universal_CA_I.pem group=sys
149 file path=etc/certs/CA/TDC_Internet_Root_CA.pem group=sys
150 file path=etc/certs/CA/TDC_OCES_Root_CA.pem group=sys
151 file path=etc/certs/CA/TURKTRUST_Certificate_Services_Provider_Root_1.pem \
152 group=sys
153 file path=etc/certs/CA/TURKTRUST_Certificate_Services_Provider_Root_2.pem \
154 group=sys
155 file path=etc/certs/CA/Taiwan_GRCA.pem group=sys
156 file path=etc/certs/CA/Thawte_Personal_Basic_CA.pem group=sys
157 file path=etc/certs/CA/Thawte_Personal_Freemail_CA.pem group=sys
158 file path=etc/certs/CA/Thawte_Personal_Premium_CA.pem group=sys
159 file path=etc/certs/CA/Thawte_Premium_Server_CA.pem group=sys
160 file path=etc/certs/CA/Thawte_Server_CA.pem group=sys
161 file path=etc/certs/CA/Thawte_Time_Stamping_CA.pem group=sys
162 file path=etc/certs/CA/UTN-USER_First-Network_Applications.pem group=sys
163 file path=etc/certs/CA/UTN_DATACorp_SGC_Root_CA.pem group=sys
164 file path=etc/certs/CA/UTN_USERFirst_Email_Root_CA.pem group=sys
165 file path=etc/certs/CA/UTN_USERFirst_Hardware_Root_CA.pem group=sys
166 file path=etc/certs/CA/UTN_USERFirst_Object_Root_CA.pem group=sys
167 file path=etc/certs/CA/ValiCert_Class_1_VA.pem group=sys
168 file path=etc/certs/CA/ValiCert_Class_2_VA.pem group=sys
169 file \
170 path=etc/certs/CA/VeriSign_Class_3_Public_Primary_Certification_Authority_-_
171 group=sys
172 file \
173 path=etc/certs/CA/Verisign_Class_1_Public_Primary_Certification_Authority.pe
174 group=sys
175 file \
176 path=etc/certs/CA/Verisign_Class_1_Public_Primary_Certification_Authority_-_
177 group=sys
178 file \
179 path=etc/certs/CA/Verisign_Class_1_Public_Primary_Certification_Authority_-_
180 group=sys
181 file \
182 path=etc/certs/CA/Verisign_Class_2_Public_Primary_Certification_Authority.pe
183 group=sys
184 file \
185 path=etc/certs/CA/Verisign_Class_2_Public_Primary_Certification_Authority_-_
186 group=sys
187 file \
188 path=etc/certs/CA/Verisign_Class_2_Public_Primary_Certification_Authority_-_
189 group=sys
190 file \
191 path=etc/certs/CA/Verisign_Class_3_Public_Primary_Certification_Authority.pe
192 group=sys
193 file \
```

```
194 path=etc/certs/CA/Verisign_Class_3_Public_Primary_Certification_Authority_-_
195 group=sys
196 file \
197 path=etc/certs/CA/Verisign_Class_3_Public_Primary_Certification_Authority_-_
198 group=sys
199 file \
200 path=etc/certs/CA/Verisign_Class_4_Public_Primary_Certification_Authority_-_
201 group=sys
202 file \
203 path=etc/certs/CA/Verisign_Class_4_Public_Primary_Certification_Authority_-_
204 group=sys
205 file path=etc/certs/CA/Verisign_RSA_Secure_Server_CA.pem group=sys
206 file path=etc/certs/CA/Verisign_Time_Stamping_Authority_CA.pem group=sys
207 file path=etc/certs/CA/Visa_International_Global_Root_2.pem group=sys
208 file path=etc/certs/CA/Visa_eCommerce_Root.pem group=sys
209 file path=etc/certs/CA/WellsSecure_Public_Root_Certificate_Authority.pem \
210 group=sys
211 file path=etc/certs/CA/Wells_Fargo_Root_CA.pem group=sys
212 file path=etc/certs/CA/XRamp_Global_CA_Root.pem group=sys
213 file path=etc/certs/CA/beTRUSTed_Root_CA-Baltimore_Implementation.pem \
214 group=sys
215 file path=etc/certs/CA/beTRUSTed_Root_CA.pem group=sys
216 file path=etc/certs/CA/beTRUSTed_Root_CA_-_Entrust_Implementation.pem \
217 group=sys
218 file path=etc/certs/CA/beTRUSTed_Root_CA_-_RSA_Implementation.pem group=sys
219 file path=etc/certs/CA/ePKI_Root_Certification_Authority.pem group=sys
220 file path=etc/certs/CA/thawte_Primary_Root_CA.pem group=sys
221 file path=lib/svc/manifest/system/certsdir.xml group=sys mode=0444
222 file path=usr/lib/certsdir mode=0555
223 license cr_Sun license=cr_Sun
224 license lic_CDDL license=lic_CDDL
225 link path=etc/openssl/certs/00673b5b.0 \
226 target=../etc/certs/CA/thawte_Primary_Root_CA.pem
227 $(OPENSSL10_ONLY)link path=etc/openssl/certs/024dc131.0 \
228 target=../etc/certs/CA/Microsec_e-Szigno_Root_CA.pem
229 link path=etc/openssl/certs/02b73561.0 \
230 target=../etc/certs/CA/Comodo_Secure_Services_root.pem
231 $(OPENSSL10_ONLY)link path=etc/openssl/certs/039c618a.0 \
232 target=../etc/certs/CA/TURKTRUST_Certificate_Services_Provider_Root_2.pem
233 $(OPENSSL10_ONLY)link path=etc/openssl/certs/03f0efa4.0 \
234 target=../etc/certs/CA/Wells_Fargo_Root_CA.pem
235 link path=etc/openssl/certs/0481cb65.0 \
236 target=../etc/certs/CA/AOL_Time_Warner_Root_Certification_Authority_2.pem
237 link path=etc/openssl/certs/052e396b.0 \
238 target=../etc/certs/CA/AddTrust_Qualified_Certificates_Root.pem
239 $(OPENSSL10_ONLY)link path=etc/openssl/certs/0750887b.0 \
240 target=../etc/certs/CA/IPS_CLASE3_root.pem
241 $(OPENSSL10_ONLY)link path=etc/openssl/certs/080911ac.0 \
242 target=../etc/certs/CA/QuoVadis_Root_CA.pem
243 link path=etc/openssl/certs/08aef7bb.0 \
244 target=../etc/certs/CA/WellsSecure_Public_Root_Certificate_Authority.pem
245 $(OPENSSL10_ONLY)link path=etc/openssl/certs/09ca81a7.0 \
246 target=../etc/certs/CA/Thawte_Personal_Premium_CA.pem
247 $(OPENSSL10_ONLY)link path=etc/openssl/certs/0a8f0c78.0 \
248 target=../etc/certs/CA/IPS_Chained_CAs_root.pem
249 link path=etc/openssl/certs/0c364b2d.0 \
250 target=../etc/certs/CA/Entrust.net_Secure_Personal_CA.pem
251 $(OPENSSL10_ONLY)link path=etc/openssl/certs/0d1b923b.0 \
252 target=../etc/certs/CA/S-TRUST_Authentication_and_Encryption_Root_CA_2005_PN.
253 link path=etc/openssl/certs/0dbd0096.0 \
254 target=../etc/certs/CA/AOL_Time_Warner_Root_Certification_Authority_1.pem
255 link path=etc/openssl/certs/0e82f83a.0 \
256 target=../etc/certs/CA/Thawte_Personal_Basic_CA.pem
257 $(OPENSSL10_ONLY)link path=etc/openssl/certs/0f11b315.0 \
258 target=../etc/certs/CA/Verisign_Class_2_Public_Primary_Certification_Authorit
259 $(OPENSSL10_ONLY)link path=etc/openssl/certs/106cd822.0 \
```



```
251 target=../certs/CA/Verisign_RSA_Secure_Server_CA.pem
252 link path=etc/openssl/certs/111e6273.0 \
253 target=../certs/CA/GlobalSign_Root_CA_-_R2.pem
254 link path=etc/openssl/certs/1155c94b.0 \
255 target=../certs/CA/Firmaprofesional_Root_CA.pem
256 link path=etc/openssl/certs/11a09b38.0 \
257 target=../certs/CA/TDC_OCES_Root_CA.pem
258 $(OPENSSL10_ONLY)link path=etc/openssl/certs/11c69ce5.0 \
259 target=../certs/CA/RSA_Security_1024_v3.pem
260 link path=etc/openssl/certs/11f154d6.0 \
261 target=../certs/CA/Verisign_Class_1_Public_Primary_Certification_Authorit
262 link path=etc/openssl/certs/124bbd54.0 \
263 target=../certs/CA/Comodo_Trusted_Services_root.pem
264 link path=etc/openssl/certs/128b9c8d.0 \
265 target=../certs/CA/Digital_Signature_Trust_Co._Global_CA_2.pem
266 $(OPENSSL10_ONLY)link path=etc/openssl/certs/12ac4d91.0 \
267 target=../certs/CA/AOL_Time_Warner_Root_Certification_Authority_2.pem
268 link path=etc/openssl/certs/12d55845.0 \
269 target=../certs/CA/DST_Root_CA_X3.pem
270 $(OPENSSL10_ONLY)link path=etc/openssl/certs/157753a5.0 \
271 target=../certs/CA/AddTrust_External_Root.pem
272 $(OPENSSL10_ONLY)link path=etc/openssl/certs/166851b2.0 \
273 target=../certs/CA/Entrust.net_Secure_Personal_CA.pem
274 $(OPENSSL10_ONLY)link path=etc/openssl/certs/16b3fe3c.0 \
275 target=../certs/CA/beTRUSTed_Root_CA_-_RSA_Implementation.pem
276 link path=etc/openssl/certs/17b51fe6.0 \
277 target=../certs/CA/Certplus_Class_2_Primary_CA.pem
278 link path=etc/openssl/certs/19899da5.0 \
279 target=../certs/CA/Entrust.net_Global_Secure_Server_CA.pem
280 link path=etc/openssl/certs/1dac3003.0 \
281 target=../certs/CA/DST_ACES_CA_X6.pem
282 link path=etc/openssl/certs/1dcd6f4c.0 target=../certs/CA/Taiwan_GRCA.pem
283 $(OPENSSL10_ONLY)link path=etc/openssl/certs/1ec4d31a.0 \
284 target=../certs/CA/Verisign_Class_3_Public_Primary_Certification_Authorit
285 $(OPENSSL10_ONLY)link path=etc/openssl/certs/201cada0.0 \
286 target=../certs/CA/America_Online_Root_Certification_Authority_2.pem
287 $(OPENSSL10_ONLY)link path=etc/openssl/certs/20d096ba.0 \
288 target=../certs/CA/ValiCert_Class_1_VA.pem
289 link path=etc/openssl/certs/219d9499.0 \
290 target=../certs/CA/Go_Daddy_Class_2_CA.pem
291 $(OPENSSL10_ONLY)link path=etc/openssl/certs/2251b13a.0 \
292 target=../certs/CA/ComSign_Secured_CA.pem
293 link path=etc/openssl/certs/23f4c490.0 \
294 target=../certs/CA/Starfield_Class_2_CA.pem
295 $(OPENSSL10_ONLY)link path=etc/openssl/certs/244b5494.0 \
296 target=../certs/CA/DigiCert_High_Assurance_EV_Root_CA.pem
297 $(OPENSSL10_ONLY)link path=etc/openssl/certs/24ad0b63.0 \
298 target=../certs/CA/Verisign_Class_1_Public_Primary_Certification_Authorit
299 link path=etc/openssl/certs/256fd83b.0 \
300 target=../certs/CA/TC_TrustCenter_Germany_Class_2_CA.pem
301 $(OPENSSL10_ONLY)link path=etc/openssl/certs/2835ab7b.0 \
302 target=../certs/CA/GTE_CyberTrust_Root_CA.pem
303 $(OPENSSL10_ONLY)link path=etc/openssl/certs/2ab3b959.0 \
304 target=../certs/CA/NetLock_Express_Class_C_Root.pem
305 link path=etc/openssl/certs/2afc57aa.0 \
306 target=../certs/CA/TC_TrustCenter_Class_2_CA_II.pem
307 $(OPENSSL10_ONLY)link path=etc/openssl/certs/2c3e3f84.0 \
308 target=../certs/CA/UTN_USERFirst_Object_Root_CA.pem
309 $(OPENSSL10_ONLY)link path=etc/openssl/certs/2c543cd1.0 \
310 target=../certs/CA/GeoTrust_Global_CA.pem
311 $(OPENSSL10_ONLY)link path=etc/openssl/certs/2cfc4974.0 \
312 target=../certs/CA/TDC_OCES_Root_CA.pem
313 $(OPENSSL10_ONLY)link path=etc/openssl/certs/2e4eed3c.0 \
314 target=../certs/CA/thawte_Primary_Root_CA.pem
315 $(OPENSSL10_ONLY)link path=etc/openssl/certs/2e5ac55d.0 \
316 target=../certs/CA/DST_Root_CA_X3.pem
```

```
317 link path=etc/openssl/certs/2edf7016.0 \
318 target=../certs/CA/Verisign_Class_1_Public_Primary_Certification_Authorit
319 link path=etc/openssl/certs/2fa87019.0 \
320 target=../certs/CA/Network_Solutions_Certificate_Authority.pem
321 link path=etc/openssl/certs/2fb1850a.0 \
322 target=../certs/CA/America_Online_Root_Certification_Authority_2.pem
323 link path=etc/openssl/certs/31044350.0 \
324 target=../certs/CA/beTRUSTed_Root_CA.pem
325 link path=etc/openssl/certs/33815e15.0 \
326 target=../certs/CA/StartCom_Certification_Authority.pem
327 link path=etc/openssl/certs/343eb6cb.0 \
328 target=../certs/CA/Cybertrust_Global_Root.pem
329 $(OPENSSL10_ONLY)link path=etc/openssl/certs/3513523f.0 \
330 target=../certs/CA/DigiCert_Global_Root_CA.pem
331 link path=etc/openssl/certs/399e7759.0 \
332 target=../certs/CA/DigiCert_Global_Root_CA.pem
333 link path=etc/openssl/certs/3a3b02ce.0 \
334 target=../certs/CA/OISTE_WISKey_Global_Root_GA_CA.pem
335 $(OPENSSL10_ONLY)link path=etc/openssl/certs/3a7f6b22.0 \
336 target=../certs/CA/Thawte_Personal_Basic_CA.pem
337 link path=etc/openssl/certs/3ad48a91.0 \
338 target=../certs/CA/Baltimore_CyberTrust_Root.pem
339 link path=etc/openssl/certs/3c58f906.0 \
340 target=../certs/CA/AddTrust_External_Root.pem
341 link path=etc/openssl/certs/3c860d51.0 \
342 target=../certs/CA/SwissSign_Gold_CA_-_G2.pem
343 $(OPENSSL10_ONLY)link path=etc/openssl/certs/3dfd7537.0 \
344 target=../certs/CA/Visa_International_Global_Root_2.pem
345 link path=etc/openssl/certs/3e7271e8.0 \
346 target=../certs/CA/Entrust.net_Premium_2048_Secure_Server_CA.pem
347 $(OPENSSL10_ONLY)link path=etc/openssl/certs/3ee7e181.0 \
348 target=../certs/CA/IGC_A.pem
349 $(OPENSSL10_ONLY)link path=etc/openssl/certs/40547a79.0 \
350 target=../certs/CA/COMODO_Certification_Authority.pem
351 $(OPENSSL10_ONLY)link path=etc/openssl/certs/40e67a49.0 \
352 target=../certs/CA/Digital_Signature_Trust_Co._Global_CA_4.pem
353 link path=etc/openssl/certs/412bea73.0 \
354 target=../certs/CA/Thawte_Personal_Premium_CA.pem
355 $(OPENSSL10_ONLY)link path=etc/openssl/certs/415660c1.0 \
356 target=../certs/CA/Verisign_Class_3_Public_Primary_Certification_Authorit
357 link path=etc/openssl/certs/4166ec0c.0 \
358 target=../certs/CA/RSA_Security_1024_v3.pem
359 link path=etc/openssl/certs/4184de39.0 \
360 target=../certs/CA/IPS_Timestamping_root.pem
361 $(OPENSSL10_ONLY)link path=etc/openssl/certs/4304c5e5.0 \
362 target=../certs/CA/Network_Solutions_Certificate_Authority.pem
363 $(OPENSSL10_ONLY)link path=etc/openssl/certs/442adcac.0 \
364 target=../certs/CA/Certum_Root_CA.pem
365 $(OPENSSL10_ONLY)link path=etc/openssl/certs/4597689c.0 \
366 target=../certs/CA/Equifax_Secure_eBusiness_CA_2.pem
367 $(OPENSSL10_ONLY)link path=etc/openssl/certs/4615970e.0 \
368 target=../certs/CA/Digital_Signature_Trust_Co._Global_CA_2.pem
369 link path=etc/openssl/certs/4643210f.0 \
370 target=../certs/CA/Digital_Signature_Trust_Co._Global_CA_4.pem
371 link path=etc/openssl/certs/46b2fd3b.0 \
372 target=../certs/CA/SwissSign_Platinum_CA_-_G2.pem
373 link path=etc/openssl/certs/47996b5c.0 \
374 target=../certs/CA/beTRUSTed_Root_CA_-_Entrust_Implementation.pem
375 $(OPENSSL10_ONLY)link path=etc/openssl/certs/480720ec.0 \
376 target=../certs/CA/GeoTrust_Primary_Certification_Authority.pem
377 $(OPENSSL10_ONLY)link path=etc/openssl/certs/48ef30f1.0 \
378 target=../certs/CA/TC_TrustCenter_Germany_Class_2_CA.pem
379 $(OPENSSL10_ONLY)link path=etc/openssl/certs/4a6481c9.0 \
380 target=../certs/CA/GlobalSign_Root_CA_-_R2.pem
381 link path=etc/openssl/certs/4d654d1d.0 \
382 target=../certs/CA/GTE_CyberTrust_Global_Root.pem
```

```
383 link path=etc/openssl/certs/4e18c148.0 \
384     target=../../certs/CA/Deutsche_Telekom_Root_CA_2.pem
385 $(OPENSSL10_ONLY)link path=etc/openssl/certs/4efc7a23.0 \
386     target=../../certs/CA/Entrust_net_Global_Secure_Server_CA.pem
387 $(OPENSSL10_ONLY)link path=etc/openssl/certs/4f316efb.0 \
388     target=../../certs/CA/SwissSign_Gold_CA_-_G2.pem
389 link path=etc/openssl/certs/4fbd6bfa.0 \
390     target=../../certs/CA/UTN_DATACorp_SGC_Root_CA.pem
391 link path=etc/openssl/certs/5021a0a2.0 \
392     target=../../certs/CA/TC_TrustCenter_Universal_CA_I.pem
393 link path=etc/openssl/certs/5046c355.0 \
394     target=../../certs/CA/SwissSign_Silver_CA_-_G2.pem
395 link path=etc/openssl/certs/54edfa5d.0 \
396     target=../../certs/CA/TC_TrustCenter_Germany_Class_3_CA.pem
397 $(OPENSSL10_ONLY)link path=etc/openssl/certs/55a10908.0 \
398     target=../../certs/CA/ValiCert_Class_2_VA.pem
399 $(OPENSSL10_ONLY)link path=etc/openssl/certs/5620c4aa.0 \
400     target=../../certs/CA/TC_TrustCenter_Class_3_CA_II.pem
401 $(OPENSSL10_ONLY)link path=etc/openssl/certs/56657bde.0 \
402     target=../../certs/CA/Comodo_Trusted_Services_root.pem
403 link path=etc/openssl/certs/56b8a0b6.0 \
404     target=../../certs/CA/TURKTRUST_Certificate_Services_Provider_Root_2.pem
405 link path=etc/openssl/certs/57692373.0 \
406     target=../../certs/CA/GeoTrust_Global_CA_2.pem
407 $(OPENSSL10_ONLY)link path=etc/openssl/certs/578d5c04.0 \
408     target=../../certs/CA/Equifax_Secure_CA.pem
409 $(OPENSSL10_ONLY)link path=etc/openssl/certs/57b0f75e.0 \
410     target=../../certs/CA/UTN-USER_First-Network_Applications.pem
411 $(OPENSSL10_ONLY)link path=etc/openssl/certs/57bcb2da.0 \
412     target=../../certs/CA/SwissSign_Silver_CA_-_G2.pem
413 link path=etc/openssl/certs/58a44af1.0 target=../../certs/CA/IGC_A.pem
414 link path=etc/openssl/certs/594f1775.0 \
415     target=../../certs/CA/Equifax_Secure_CA.pem
416 link path=etc/openssl/certs/5a3f0ff8.0 \
417     target=../../certs/CA/COMODO_Certification_Authority.pem
418 link path=etc/openssl/certs/5a5372fc.0 \
419     target=../../certs/CA/NetLock_Business_Class_B_Root.pem
420 $(OPENSSL10_ONLY)link path=etc/openssl/certs/5a950642.0 \
421     target=../../certs/CA/IPS_CLASEA1_root.pem
422 $(OPENSSL10_ONLY)link path=etc/openssl/certs/5ad8a5d6.0 \
423     target=../../certs/CA/GlobalSign_Root_CA.pem
424 link path=etc/openssl/certs/5cf9d536.0 \
425     target=../../certs/CA/QuoVadis_Root_CA.pem
426 $(OPENSSL10_ONLY)link path=etc/openssl/certs/5f267794.0 \
427     target=../../certs/CA/Entrust_net_Secure_Server_CA.pem
428 link path=etc/openssl/certs/635ccfd5.0 \
429     target=../../certs/CA/NetLock_Express_Class_C_Root.pem
430 $(OPENSSL10_ONLY)link path=etc/openssl/certs/6410666e.0 \
431     target=../../certs/CA/Taiwan_GRCA.pem
432 $(OPENSSL10_ONLY)link path=etc/openssl/certs/64d1f6f4.0 \
433     target=../../certs/CA/Thawte_Personal_Freemail_CA.pem
434 $(OPENSSL10_ONLY)link path=etc/openssl/certs/653b494a.0 \
435     target=../../certs/CA/Baltimore_CyberTrust_Root.pem
436 $(OPENSSL10_ONLY)link path=etc/openssl/certs/667c66d4.0 \
437     target=../../certs/CA/Swisscom_Root_CA_1.pem
438 $(OPENSSL10_ONLY)link path=etc/openssl/certs/67d559d1.0 \
439     target=../../certs/CA/Sonera_Class_1_Root_CA.pem
440 $(OPENSSL10_ONLY)link path=etc/openssl/certs/686ef281.0 \
441     target=../../certs/CA/beTRUSTed_Root_CA_-_Entrust_Implementation.pem
442 link path=etc/openssl/certs/69105f4f.0 \
443     target=../../certs/CA/DigiCert_Assured_ID_Root_CA.pem
444 link path=etc/openssl/certs/6adf0799.0 \
445     target=../../certs/CA/Wells_Fargo_Root_CA.pem
446 $(OPENSSL10_ONLY)link path=etc/openssl/certs/6b99d060.0 \
447     target=../../certs/CA/Entrust_Root_Certification_Authority.pem
448 $(OPENSSL10_ONLY)link path=etc/openssl/certs/6cb3815b.0 \
```

```
449     target=../../certs/CA/Verisign_Time_Stamping_Authority_CA.pem
450 $(OPENSSL10_ONLY)link path=etc/openssl/certs/6cc3c4c3.0 \
451     target=../../certs/CA/Thawte_Server_CA.pem
452 $(OPENSSL10_ONLY)link path=etc/openssl/certs/6db5a580.0 \
453     target=../../certs/CA/IPS_CLASEA3_root.pem
454 link path=etc/openssl/certs/6e8bf996.0 \
455     target=../../certs/CA/Certum_Root_CA.pem
456 $(OPENSSL10_ONLY)link path=etc/openssl/certs/6faac4e3.0 \
457     target=../../certs/CA/Verisign_Class_4_Public_Primary_Certification_Authorit
458 link path=etc/openssl/certs/6fcc125d.0 \
459     target=../../certs/CA/Visa_eCommerce_Root.pem
460 $(OPENSSL10_ONLY)link path=etc/openssl/certs/706f604c.0 \
461     target=../../certs/CA/XRamp_Global_CA_Root.pem
462 link path=etc/openssl/certs/709afd2b.0 \
463     target=../../certs/CA/Thawte_Personal_Freemail_CA.pem
464 link path=etc/openssl/certs/72bf6a04.0 \
465     target=../../certs/CA/IPS_CLASEA3_root.pem
466 link path=etc/openssl/certs/72fa7371.0 \
467     target=../../certs/CA/Verisign_Class_3_Public_Primary_Certification_Authorit
468 link path=etc/openssl/certs/74c26bd0.0 \
469     target=../../certs/CA/Equifax_Secure_Global_eBusiness_CA.pem
470 link path=etc/openssl/certs/755f7420.0 \
471     target=../../certs/CA/Sonera_Class_1_Root_CA.pem
472 link path=etc/openssl/certs/75680d2e.0 \
473     target=../../certs/CA/Comodo_AAA_Services_root.pem
474 link path=etc/openssl/certs/7651b327.0 \
475     target=../../certs/CA/Verisign_Class_3_Public_Primary_Certification_Authorit
476 link path=etc/openssl/certs/76579174.0 \
477     target=../../certs/CA/XRamp_Global_CA_Root.pem
478 $(OPENSSL10_ONLY)link path=etc/openssl/certs/76c8bf92.0 \
479     target=../../certs/CA/Cybertrust_Global_Root.pem
480 $(OPENSSL10_ONLY)link path=etc/openssl/certs/76faf6c0.0 \
481     target=../../certs/CA/QuoVadis_Root_CA_3.pem
482 $(OPENSSL10_ONLY)link path=etc/openssl/certs/778e3cb0.0 \
483     target=../../certs/CA/UTN_DATACorp_SGC_Root_CA.pem
484 link path=etc/openssl/certs/788c9bfc.0 \
485     target=../../certs/CA/Visa_International_Global_Root_2.pem
486 $(OPENSSL10_ONLY)link path=etc/openssl/certs/790a7190.0 \
487     target=../../certs/CA/DST_ACES_CA_X6.pem
488 link path=etc/openssl/certs/7999be0d.0 \
489     target=../../certs/CA/GeoTrust_Global_CA.pem
490 $(OPENSSL10_ONLY)link path=etc/openssl/certs/79ad8b43.0 \
491     target=../../certs/CA/Equifax_Secure_eBusiness_CA_1.pem
492 link path=etc/openssl/certs/7a481e66.0 \
493     target=../../certs/CA/TC_TrustCenter_Class_3_CA_II.pem
494 link path=etc/openssl/certs/7a819ef2.0 \
495     target=../../certs/CA/QuoVadis_Root_CA_2.pem
496 link path=etc/openssl/certs/7d3cd826.0 \
497     target=../../certs/CA/RSA_Root_Certificate_1.pem
498 link path=etc/openssl/certs/7d453d8f.0 \
499     target=../../certs/CA/Verisign_Class_3_Public_Primary_Certification_Authorit
500 $(OPENSSL10_ONLY)link path=etc/openssl/certs/7d5a75e4.0 \
501     target=../../certs/CA/WellsSecure_Public_Root_Certificate_Authority.pem
502 $(OPENSSL10_ONLY)link path=etc/openssl/certs/812e17de.0 \
503     target=../../certs/CA/Deutsche_Telekom_Root_CA_2.pem
504 link path=etc/openssl/certs/81b9768f.0 \
505     target=../../certs/CA/DigiCert_High_Assurance_EV_Root_CA.pem
506 link path=etc/openssl/certs/8317b10c.0 \
507     target=../../certs/CA/NetLock_Qualified_Class_QA_Root.pem
508 link path=etc/openssl/certs/8470719d.0 \
509     target=../../certs/CA/RSA_Security_2048_v3.pem
510 link path=etc/openssl/certs/84cba82f.0 \
511     target=../../certs/CA/TURKTRUST_Certificate_Services_Provider_Root_1.pem
512 $(OPENSSL10_ONLY)link path=etc/openssl/certs/861a399d.0 \
513     target=../../certs/CA/AddTrust_Low-Value_Services_Root.pem
514 $(OPENSSL10_ONLY)link path=etc/openssl/certs/861e0100.0 \
```

```
515 target=../certs/CA/NetLock_Qualified_Class_QA_Root.pem
516 link path=etc/openssl/certs/86f32474.0 \
517 target=../certs/CA/IPS_CLASE3_root.pem
518 link path=etc/openssl/certs/87753b0d.0 \
519 target=../certs/CA/GeoTrust_Universal_CA_2.pem
520 $(OPENSSL10_ONLY)link path=etc/openssl/certs/8867006a.0 \
521 target=../certs/CA/GeoTrust_Universal_CA_2.pem
522 $(OPENSSL10_ONLY)link path=etc/openssl/certs/88f89ea7.0 \
523 target=../certs/CA/TURKTRUST_Certificate_Services_Provider_Root_1.pem
524 link path=etc/openssl/certs/89c02a45.0 \
525 target=../certs/CA/COMODO_ECC_Certification_Authority.pem
526 $(OPENSSL10_ONLY)link path=etc/openssl/certs/8b59blad.0 \
527 target=../certs/CA/AddTrust_Public_Services_Root.pem
528 $(OPENSSL10_ONLY)link path=etc/openssl/certs/8f111d69.0 \
529 target=../certs/CA/MD5_Collisions_Forged_Rogue_CA_25c3.pem
530 link path=etc/openssl/certs/8f7b96c4.0 \
531 target=../certs/CA/Equifax_Secure_eBusiness_CA_2.pem
532 link path=etc/openssl/certs/8fe643df.0 \
533 target=../certs/CA/IPS_CLASEA1_root.pem
534 link path=etc/openssl/certs/9339512a.0 \
535 target=../certs/CA/QuoVadis_Root_CA_3.pem
536 link path=etc/openssl/certs/95750816.0 \
537 target=../certs/CA/Verisign_Time_Stamping_Authority_CA.pem
538 $(OPENSSL10_ONLY)link path=etc/openssl/certs/972672fc.0 \
539 target=../certs/CA/TC_TrustCenter_Germany_Class_3_CA.pem
540 link path=etc/openssl/certs/9772ca32.0 \
541 target=../certs/CA/GeoTrust_Primary_Certification_Authority.pem
542 link path=etc/openssl/certs/97b4211c.0 \
543 target=../certs/CA/GTE_CyberTrust_Root_CA.pem
544 $(OPENSSL10_ONLY)link path=etc/openssl/certs/98ec67f0.0 \
545 target=../certs/CA/Thawte_Premium_Server_CA.pem
546 $(OPENSSL10_ONLY)link path=etc/openssl/certs/9af9f759.0 \
547 target=../certs/CA/RSA_Root_Certificate_1.pem
548 $(OPENSSL10_ONLY)link path=etc/openssl/certs/9b353c9a.0 \
549 target=../certs/CA/TDC_Internet_Root_CA.pem
550 $(OPENSSL10_ONLY)link path=etc/openssl/certs/9c2e7d30.0 \
551 target=../certs/CA/Sonera_Class_2_Root_CA.pem
552 $(OPENSSL10_ONLY)link path=etc/openssl/certs/9d520b32.0 \
553 target=../certs/CA/Security_Communication_EV_RootCA1.pem
554 link path=etc/openssl/certs/9d6523ce.0 \
555 target=../certs/CA/ePKI_Root_Certification_Authority.pem
556 link path=etc/openssl/certs/9dbefe7b.0 \
557 target=../certs/CA/Security_Communication_EV_RootCA1.pem
558 $(OPENSSL10_ONLY)link path=etc/openssl/certs/9e6afd31.0 \
559 target=../certs/CA/Thawte_Time_Stamping_CA.pem
560 link path=etc/openssl/certs/9ec3a561.0 \
561 target=../certs/CA/UTN_USERFirst_Email_Root_CA.pem
562 $(OPENSSL10_ONLY)link path=etc/openssl/certs/9f541fb4.0 \
563 target=../certs/CA/Digital_Signature_Trust_Co_Global_CA_3.pem
564 $(OPENSSL10_ONLY)link path=etc/openssl/certs/a01d1cc2.0 \
565 target=../certs/CA/IPS_Servidores_root.pem
566 link path=etc/openssl/certs/a0bc6fbb.0 \
567 target=../certs/CA/Camerfirma_Global_Chambersign_Root.pem
568 $(OPENSSL10_ONLY)link path=etc/openssl/certs/a137bd1c.0 \
569 target=../certs/CA/beTRUSTed_Root_CA-Baltimore_Implementation.pem
570 link path=etc/openssl/certs/a15b3b6b.0 \
571 target=../certs/CA/Digital_Signature_Trust_Co_Global_CA_3.pem
572 link path=etc/openssl/certs/a2df7ad7.0 \
573 target=../certs/CA/AddTrust_Public_Services_Root.pem
574 link path=etc/openssl/certs/a3896b44.0 \
575 target=../certs/CA/Security_Communication_Root_CA.pem
576 $(OPENSSL10_ONLY)link path=etc/openssl/certs/a5fd78f0.0 \
577 target=../certs/CA/TC_TrustCenter_Class_2_CA_II.pem
578 link path=etc/openssl/certs/a6776c69.0 \
579 target=../certs/CA/IPS_Chained_CAs_root.pem
580 $(OPENSSL10_ONLY)link path=etc/openssl/certs/a6a593ba.0 \
```

```
581 target=../certs/CA/Digital_Signature_Trust_Co_Global_CA_1.pem
582 link path=etc/openssl/certs/a7605362.0 \
583 target=../certs/CA/Sonera_Class_2_Root_CA.pem
584 $(OPENSSL10_ONLY)link path=etc/openssl/certs/a760e1bd.0 \
585 target=../certs/CA/Visa_eCommerce_Root.pem
586 $(OPENSSL10_ONLY)link path=etc/openssl/certs/a8dee976.0 \
587 target=../certs/CA/SwissSign_Platinum_CA_-_G2.pem
588 link path=etc/openssl/certs/aaa45464.0 \
589 target=../certs/CA/Thawte_Time_Stamping_CA.pem
590 $(OPENSSL10_ONLY)link path=etc/openssl/certs/ad088e1d.0 \
591 target=../certs/CA/GeoTrust_Universal_CA.pem
592 link path=etc/openssl/certs/add67345.0 \
593 target=../certs/CA/NetLock_Notary_Class_A_Root.pem
594 $(OPENSSL10_ONLY)link path=etc/openssl/certs/ae8153b9.0 \
595 target=../certs/CA/StartCom_Certification_Authority.pem
596 $(OPENSSL10_ONLY)link path=etc/openssl/certs/aeef10d.0 \
597 target=../certs/CA/Entrust.net_Premium_2048_Secure_Server_CA.pem
598 $(OPENSSL10_ONLY)link path=etc/openssl/certs/af4f0c93.0 \
599 target=../certs/CA/Entrust.net_Global_Secure_Personal_CA.pem
600 link path=etc/openssl/certs/b0f3e76e.0 \
601 target=../certs/CA/GlobalSign_Root_CA.pem
602 $(OPENSSL10_ONLY)link path=etc/openssl/certs/b1159c4c.0 \
603 target=../certs/CA/DigiCert_Assured_ID_Root_CA.pem
604 $(OPENSSL10_ONLY)link path=etc/openssl/certs/b13cc6df.0 \
605 target=../certs/CA/UTN_USERFirst_Hardware_Root_CA.pem
606 $(OPENSSL10_ONLY)link path=etc/openssl/certs/b1b8a7f3.0 \
607 target=../certs/CA/OISTE_WISKey_Global_Root_CA_CA.pem
608 $(OPENSSL10_ONLY)link path=etc/openssl/certs/b204d74a.0 \
609 target=../certs/CA/VeriSign_Class_3_Public_Primary_Certification_Authorit
610 $(OPENSSL10_ONLY)link path=etc/openssl/certs/b42ff584.0 \
611 target=../certs/CA/Staat_der_Nederlanden_Root_CA.pem
612 link path=etc/openssl/certs/b5f329fa.0 \
613 target=../certs/CA/Verisign_Class_2_Public_Primary_Certification_Authorit
614 $(OPENSSL10_ONLY)link path=etc/openssl/certs/b66938e9.0 \
615 target=../certs/CA/Secure_Global_CA.pem
616 $(OPENSSL10_ONLY)link path=etc/openssl/certs/b7e7231a.0 \
617 target=../certs/CA/NetLock_Business_Class_B_Root.pem
618 $(OPENSSL10_ONLY)link path=etc/openssl/certs/b8e83700.0 \
619 target=../certs/CA/Verisign_Class_1_Public_Primary_Certification_Authorit
620 $(OPENSSL10_ONLY)link path=etc/openssl/certs/bad35b78.0 \
621 target=../certs/CA/Verisign_Class_4_Public_Primary_Certification_Authorit
622 $(OPENSSL10_ONLY)link path=etc/openssl/certs/bb2d49a0.0 \
623 target=../certs/CA/ComSign_CA.pem
624 link path=etc/openssl/certs/bcdd5959.0 \
625 target=../certs/CA/ValiCert_Class_2_VA.pem
626 link path=etc/openssl/certs/bda4cc84.0 \
627 target=../certs/CA/America_Online_Root_Certification_Authority_1.pem
628 link path=etc/openssl/certs/bdacc6f.0 \
629 target=../certs/CA/Secure_Global_CA.pem
630 link path=etc/openssl/certs/bf64f35b.0 \
631 target=../certs/CA/Entrust_Root_Certification_Authority.pem
632 link path=etc/openssl/certs/bf87590f.0 \
633 target=../certs/CA/beTRUSTed_Root_CA_-_RSA_Implementation.pem
634 $(OPENSSL10_ONLY)link path=etc/openssl/certs/c0ff1f52.0 \
635 target=../certs/CA/Verisign_Class_3_Public_Primary_Certification_Authorit
636 link path=etc/openssl/certs/c19d42c7.0 \
637 target=../certs/CA/Verisign_Class_1_Public_Primary_Certification_Authorit
638 link path=etc/openssl/certs/c215bc69.0 \
639 target=../certs/CA/Digital_Signature_Trust_Co_Global_CA_1.pem
640 link path=etc/openssl/certs/c33a80d4.0 \
641 target=../certs/CA/Thawte_Premium_Server_CA.pem
642 link path=etc/openssl/certs/c527e4ab.0 \
643 target=../certs/CA/Verisign_Class_4_Public_Primary_Certification_Authorit
644 $(OPENSSL10_ONLY)link path=etc/openssl/certs/c5e082db.0 \
645 target=../certs/CA/UTN_USERFirst_Email_Root_CA.pem
646 $(OPENSSL10_ONLY)link path=etc/openssl/certs/c692a373.0 \
```

```
647 target=../../certs/CA/GTE_CyberTrust_Global_Root.pem
648 $(OPENSSL10_ONLY)link path=etc/openssl/certs/c8841d13.0 \
649 target=../../certs/CA/TC_TrustCenter_Universal_CA_1.pem
650 $(OPENSSL10_ONLY)link path=etc/openssl/certs/c99398f3.0 \
651 target=../../certs/CA/RSA_Security_2048_v3.pem
652 link path=etc/openssl/certs/c9bc75ba.0 \
653 target=../../certs/CA/ABAecom_sub.,_Am._Bankers_Assn._Root_CA.pem
654 $(OPENSSL10_ONLY)link path=etc/openssl/certs/c9f83a1c.0 \
655 target=../../certs/CA/Comodo_Secure_Services_root.pem
656 $(OPENSSL10_ONLY)link path=etc/openssl/certs/ca6e4ad9.0 \
657 target=../../certs/CA/ePKI_Root_Certification_Authority.pem
658 $(OPENSSL10_ONLY)link path=etc/openssl/certs/cb357862.0 \
659 target=../../certs/CA/Verisign_Class_2_Public_Primary_Certification_Authorit
660 $(OPENSSL10_ONLY)link path=etc/openssl/certs/cb59f961.0 \
661 target=../../certs/CA/Camerfirma_Global_Chambersign_Root.pem
662 link path=etc/openssl/certs/cb796bc1.0 target=../../certs/CA/StartCom_Ltd..pem
663 $(OPENSSL10_ONLY)link path=etc/openssl/certs/cbeee9e2.0 \
664 target=../../certs/CA/GeoTrust_Global_CA_2.pem
665 link path=etc/openssl/certs/ccb919f9.0 \
666 target=../../certs/CA/UTN_USERFirst_Object_Root_CA.pem
667 link path=etc/openssl/certs/cdaebb72.0 \
668 target=../../certs/CA/Staat_der_Nederlanden_Root_CA.pem
669 $(OPENSSL10_ONLY)link path=etc/openssl/certs/ce026bf8.0 \
670 target=../../certs/CA/Firmaprofesional_Root_CA.pem
671 link path=etc/openssl/certs/cf701eeb.0 \
672 target=../../certs/CA/SecureTrust_CA.pem
673 $(OPENSSL10_ONLY)link path=etc/openssl/certs/d0cba2e5.0 \
674 target=../../certs/CA/StartCom_Ltd..pem
675 link path=etc/openssl/certs/d2adc77d.0 \
676 target=../../certs/CA/Entrust.net_Global_Secure_Personal_CA.pem
677 $(OPENSSL10_ONLY)link path=etc/openssl/certs/d495d385.0 \
678 target=../../certs/CA/IPS_Timestamping_root.pem
679 link path=etc/openssl/certs/d537fba6.0 \
680 target=../../certs/CA/TDC_Internet_Root_CA.pem
681 link path=etc/openssl/certs/d78a75c7.0 \
682 target=../../certs/CA/Verisign_Class_2_Public_Primary_Certification_Authorit
683 $(OPENSSL10_ONLY)link path=etc/openssl/certs/d7e8dc79.0 \
684 target=../../certs/CA/QuoVadis_Root_CA_2.pem
685 link path=etc/openssl/certs/d8274e24.0 \
686 target=../../certs/CA/UTN-USER_First-Network_Applications.pem
687 $(OPENSSL10_ONLY)link path=etc/openssl/certs/d9d12c58.0 \
688 target=../../certs/CA/NetLock_Notary_Class_A_Root.pem
689 $(OPENSSL10_ONLY)link path=etc/openssl/certs/dc45b0bd.0 \
690 target=../../certs/CA/Verisign_Class_2_Public_Primary_Certification_Authorit
691 $(OPENSSL10_ONLY)link path=etc/openssl/certs/dc4ebcb9.0 \
692 target=../../certs/CA/IPS_CLASE1_root.pem
693 link path=etc/openssl/certs/ddc328ff.0 \
694 target=../../certs/CA/Thawte_Server_CA.pem
695 $(OPENSSL10_ONLY)link path=etc/openssl/certs/e113c810.0 \
696 target=../../certs/CA/Certigna.pem
697 link path=etc/openssl/certs/e2684ac5.0 \
698 target=../../certs/CA/AddTrust_Low-Value_Services_Root.pem
699 $(OPENSSL10_ONLY)link path=etc/openssl/certs/e536d871.0 \
700 target=../../certs/CA/AddTrust_Qualified_Certificates_Root.pem
701 link path=etc/openssl/certs/e60bf0c0.0 \
702 target=../../certs/CA/Swisscom_Root_CA_1.pem
703 link path=etc/openssl/certs/e7461595.0 \
704 target=../../certs/CA/MD5_Collisions_Forged_Rogue_CA_25c3.pem
705 link path=etc/openssl/certs/e775ed2d.0 \
706 target=../../certs/CA/GeoTrust_Universal_CA.pem
707 link path=etc/openssl/certs/e7b8d656.0 \
708 target=../../certs/CA/Equifax_Secure_eBusiness_CA_1.pem
709 $(OPENSSL10_ONLY)link path=etc/openssl/certs/eacdeb40.0 \
710 target=../../certs/CA/America_Online_Root_Certification_Authority_1.pem
711 link path=etc/openssl/certs/ed049835.0 \
712 target=../../certs/CA/Verisign_Class_4_Public_Primary_Certification_Authorit
```

```
713 link path=etc/openssl/certs/ed524cf5.0 \
714 target=../../certs/CA/Entrust.net_Secure_Server_CA.pem
715 link path=etc/openssl/certs/ed62f4e3.0 \
716 target=../../certs/CA/Verisign_Class_2_Public_Primary_Certification_Authorit
717 $(OPENSSL10_ONLY)link path=etc/openssl/certs/ed9bb25c.0 \
718 target=../../certs/CA/AOL_Time_Warner_Root_Certification_Authority_1.pem
719 $(OPENSSL10_ONLY)link path=etc/openssl/certs/ee1365c0.0 \
720 target=../../certs/CA/Verisign_Class_1_Public_Primary_Certification_Authorit
721 $(OPENSSL10_ONLY)link path=etc/openssl/certs/ee64a828.0 \
722 target=../../certs/CA/Comodo_AAA_Services_root.pem
723 link path=etc/openssl/certs/ee7cd6fb.0 \
724 target=../../certs/CA/Camerfirma_Chambers_of_Commerce_Root.pem
725 $(OPENSSL10_ONLY)link path=etc/openssl/certs/eed8c118.0 \
726 target=../../certs/CA/COMODO_ECC_Certification_Authority.pem
727 $(OPENSSL10_ONLY)link path=etc/openssl/certs/ef2f636c.0 \
728 target=../../certs/CA/Equifax_Secure_Global_eBusiness_CA.pem
729 $(OPENSSL10_ONLY)link path=etc/openssl/certs/f060240e.0 \
730 target=../../certs/CA/Certplus_Class_2_Primary_CA.pem
731 $(OPENSSL10_ONLY)link path=etc/openssl/certs/f081611a.0 \
732 target=../../certs/CA/Go_Daddy_Class_2_CA.pem
733 $(OPENSSL10_ONLY)link path=etc/openssl/certs/f2cce23a.0 \
734 target=../../certs/CA/beTRUSTed_Root_CA.pem
735 $(OPENSSL10_ONLY)link path=etc/openssl/certs/f3377b1b.0 \
736 target=../../certs/CA/Security_Communication_Root_CA.pem
737 $(OPENSSL10_ONLY)link path=etc/openssl/certs/f387163d.0 \
738 target=../../certs/CA/Starfield_Class_2_CA.pem
739 $(OPENSSL10_ONLY)link path=etc/openssl/certs/f39fc864.0 \
740 target=../../certs/CA/SecureTrust_CA.pem
741 link path=etc/openssl/certs/f3cf1e8e.0 \
742 target=../../certs/CA/beTRUSTed_Root_CA-Baltimore_Implementation.pem
743 link path=etc/openssl/certs/f4996e82.0 \
744 target=../../certs/CA/ValiCert_Class_1_VA.pem
745 link path=etc/openssl/certs/f559733c.0 \
746 target=../../certs/CA/S-TRUST_Authentication_and_Encryption_Root_CA_2005_PN.
747 link path=etc/openssl/certs/f58a60fe.0 \
748 target=../../certs/CA/ComSign_Secured_CA.pem
749 link path=etc/openssl/certs/f61bff45.0 \
750 target=../../certs/CA/Microsec_e-Szigno_Root_CA.pem
751 link path=etc/openssl/certs/f64d9715.0 \
752 target=../../certs/CA/IPS_Servidores_root.pem
753 link path=etc/openssl/certs/f73e89fd.0 \
754 target=../../certs/CA/Verisign_RSA_Secure_Server_CA.pem
755 $(OPENSSL10_ONLY)link path=etc/openssl/certs/f90208f7.0 \
756 target=../../certs/CA/Camerfirma_Chambers_of_Commerce_Root.pem
757 link path=etc/openssl/certs/f950ccc2.0 \
758 target=../../certs/CA/IPS_CLASE1_root.pem
759 link path=etc/openssl/certs/facacbc6.0 \
760 target=../../certs/CA/VeriSign_Class_3_Public_Primary_Certification_Authorit
761 link path=etc/openssl/certs/fde84897.0 target=../../certs/CA/Certigna.pem
762 $(OPENSSL10_ONLY)link path=etc/openssl/certs/fe30b214.0 \
763 target=../../certs/CA/ABAecom_sub.,_Am._Bankers_Assn._Root_CA.pem
764 link path=etc/openssl/certs/ff588423.0 target=../../certs/CA/ComSign_CA.pem
765 link path=etc/openssl/certs/ff783690.0 \
766 target=../../certs/CA/UTN_USERFirst_Hardware_Root_CA.pem
```