

new/usr/src/cmd/intrd/Makefile

1

1267 Sat Sep 14 15:00:19 2013

new/usr/src/cmd/intrd/Makefile

XXX Rearchitect and replace interrupt distribution

```
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 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # Makefile for interrupt distribution daemon
27 #
```

```
29 PROG=      intrd
29 MANIFEST=   intrd.xml
30 SVCMETHOD=   svc-intrd
```

```
32 include ../Makefile.cmd
```

```
34 TARGET= all
```

```
36 ROOTMANIFESTDIR=      $(ROOTSVCSYSTEM)
```

```
38 # install macros and rule
```

```
39 #
```

```
41 ROOTPROG=      $(ROOTLIB)/$(PROG)
```

```
42 $(ROOTPROG)    := FILEMODE= 555
```

```
41 .KEEP_STATE:
```

```
43 all: $(PROG)
```

```
45 install: all .WAIT $(ROOTMANIFEST) $(ROOTSVCMETHOD)
```

```
48 install: all .WAIT $(ROOTPROG) $(ROOTMANIFEST) $(ROOTSVCMETHOD)
```

```
47 clean:
```

```
49 check: $(CHKMANIFEST)
```

```
51 ${ROOTLIB}/%: %
```

```
52     ${INS.file}
```

```
54 lint:
```

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

new/usr/src/cmd/intrd/intrd.xml

1

```
*****
2189 Sat Sep 14 15:00:22 2013
new/usr/src/cmd/intrd/intrd.xml
XXX Rearchitect and replace interrupt distribution
*****
1 <?xml version="1.0"?>
2 <!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
3 <!--
4 CDDL HEADER START

6 The contents of this file are subject to the terms of the
7 Common Development and Distribution License, Version 1.0 only
8 (the "License"). You may not use this file except in compliance
9 with the License.

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

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

22 CDDL HEADER END

24 Copyright 2005 Sun Microsystems, Inc. All rights reserved.
25 Use is subject to license terms.

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

27 NOTE: This service manifest is not editable; its contents will
28 be overwritten by package or patch operations, including
29 operating system upgrade. Make customizations in a different
30 file.
31 -->

33 <service_bundle type='manifest' name='SUNWckr:intrd'>

35 <service
36 name='system/intrd'
37 type='service'
38 version='1'>

40 <create_default_instance enabled='false' />

42 <single_instance/>

44 <dependency
45 name='milestone'
46 grouping='require_all'
47 restart_on='none'
48 type='service'>
49 <service_fmri value='svc:/milestone/multi-user' />
50 </dependency>

52 <exec_method
53 type='method'
54 name='start'
55 exec='/lib/svc/method/svc-intrd start'
56 exec='/lib/svc/method/svc-intrd'
57 timeout_seconds='60' />

58 <exec_method
```

new/usr/src/cmd/intrd/intrd.xml

2

```
59 type='method'
60 name='stop'
61 exec='/lib/svc/method/svc-intrd stop'
62 exec=':kill'
63 timeout_seconds='10' />

64 <property_group name='startd' type='framework'>
65 <propval name='duration' type='astring' value='transient' />
66 </property_group>

68 #endif /* ! codereview */
69 <stability value='Unstable' />

71 <template>
72 <common_name>
73 <loctext xml:lang='C'> interrupt balancer
74 </loctext>
75 </common_name>
76 <documentation>
77 <manpage title='intrd' section='1M'
78 manpath='/usr/share/man' />
79 </documentation>
80 </template>
81 </service>

83 </service_bundle>
```

new/usr/src/cmd/intrd/svc-intrd

1

1113 Sat Sep 14 15:00:23 2013

new/usr/src/cmd/intrd/svc-intrd

XXX Rearchitect and replace interrupt distribution

```
1 #!/sbin/sh
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License, Version 1.0 only
7 # (the "License"). You may not use this file except in compliance
8 # with the License.
9 #
10 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
11 # or http://www.opensolaris.org/os/licensing.
12 # See the License for the specific language governing permissions
13 # and limitations under the License.
14 #
15 # When distributing Covered Code, include this CDDL HEADER in each
16 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
17 # If applicable, add the following below this CDDL HEADER, with the
18 # fields enclosed by brackets "[]" replaced with your own identifying
19 # information: Portions Copyright [yyyy] [name of copyright owner]
20 #
21 # CDDL HEADER END
22 #
```

```
24 #
25 # Copyright 2005 Sun Microsystems, Inc. All rights reserved.
26 # Use is subject to license terms.
27 #
```

```
28 # ident "%Z%M% %I% %E% SMI"
```

```
29 . /lib/svc/share/smf_include.sh
```

```
31 #
32 # Main
33 #
34 case "$1" in
35 'start')
36     #/sbin/uadmin 24 1 >/dev/null 2>&1
37     ;;
38 'stop')
39     #/sbin/uadmin 24 0 >/dev/null 2>&1
40     ;;
41 esac;
32 # Don't start intrd if intrd can't be found or if intrd is already running.
```

```
43 exit $SMF_EXIT_OK
34 if [ ! -f /usr/lib/intrd ]; then
35     echo "$0: intrd daemon not installed"
36     exit $SMF_EXIT_ERR_CONFIG
37 fi
```

```
39 if /usr/bin/pgrep -x -z global intrd >/dev/null; then
40     echo "$0: intrd is already running"
41     exit $SMF_EXIT_ERR_FATAL
42 fi
```

```
44 /usr/lib/intrd &
```

new/usr/src/cmd/uadmin/uadmin.c

1

```
*****
10595 Sat Sep 14 15:00:24 2013
new/usr/src/cmd/uadmin/uadmin.c
XXX Rearchitect and replace interrupt distribution
*****

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) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
23 */

25 /*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
26 /*      All Rights Reserved */

30 #include <errno.h>
31 #include <fcntl.h>
32 #include <stdio.h>
33 #include <stdlib.h>
34 #include <string.h>
35 #include <strings.h>
36 #include <signal.h>
37 #include <unistd.h>
38 #ifdef __i386
39 #include <libscf_priv.h>
40 #endif /* __i386 */

42 #include <bsm/adt.h>
43 #include <bsm/adt_event.h>

45 #include <sys/types.h>
46 #include <sys/uadmin.h>
47 #include <sys/wait.h>

49 #define SMF_RST "/etc/svc/volatile/resetting"
50 #define RETRY_COUNT 15 /* number of 1 sec retries for audit(1M) to complete */

52 static const char *Usage = "Usage: %s cmd fcn [mdep]\n";

54 static int closeout_audit(int, int);
55 static int turnoff_auditd(void);
56 static void wait_for_aqueue();
57 static int change_audit_file(void);

59 int
60 main(int argc, char *argv[])
61 {
```

new/usr/src/cmd/uadmin/uadmin.c

2

```
62     int cmd, fcn;
63     uintptr_t mdep = NULL;
64     sigset_t set;
65     adt_session_data_t *ah; /* audit session handle */
66     adt_event_data_t *event = NULL; /* event to be generated */
67     au_event_t event_id;
68     enum adt_uadmin_fcn fcn_id;

70     if (argc < 3 || argc > 4) {
71         (void) fprintf(stderr, Usage, argv[0]);
72         return (1);
73     }

75     (void) sigfillset(&set);
76     (void) sigprocmask(SIG_BLOCK, &set, NULL);

78     cmd = atoi(argv[1]);
79     fcn = atoi(argv[2]);
80     if (argc == 4) { /* mdep argument given */
81         if (cmd != A_REBOOT && cmd != A_SHUTDOWN && cmd != A_DUMP &&
82             cmd != A_FREEZE) {
83             (void) fprintf(stderr, "%s: mdep argument not "
84                 "allowed for this cmd value\n", argv[0]);
85             (void) fprintf(stderr, Usage, argv[0]);
86             return (1);
87         } else {
88             mdep = (uintptr_t)argv[3];
89         }
90     }

92     /* set up audit session and event */
93     if (adt_start_session(&ah, NULL, ADT_USE_PROC_DATA) != 0) {
94         (void) fprintf(stderr, "%s: can't start audit session\n",
95             argv[0]);
96     }

97     switch (cmd) {
98     case A_SHUTDOWN:
99         event_id = ADT_uadmin_shutdown;
100         break;
101     case A_REBOOT:
102         event_id = ADT_uadmin_reboot;
103         break;
104     case A_DUMP:
105         event_id = ADT_uadmin_dump;
106         break;
107     case A_REMOUNT:
108         event_id = ADT_uadmin_remount;
109         break;
110     case A_FREEZE:
111         event_id = ADT_uadmin_freeze;
112         break;
113     case A_FTRACE:
114         event_id = ADT_uadmin_ftrace;
115         break;
116     case A_CONFIG:
117         event_id = ADT_uadmin_config;
118         break;
119     case A_SWAPCTL:
120         event_id = ADT_uadmin_swapctl;
121         break;
122     case A_INTRD:
123         event_id = ADT_uadmin_intrd;
124         break;
125 #endif /* ! codereview */
126     default:
127         event_id = 0;
```



```

260         event->adt_uadmin_intrd.fcn = fcn_id;
261         break;
262 #endif /* ! codereview */
263     }

265     if (adt_put_event(event, ADT_SUCCESS, 0) != 0) {
266         (void) fprintf(stderr,
267             "%s: can't put audit event\n", argv[0]);
268     }
269     /*
270      * allow audit record to be processed in the kernel
271      * audit queue
272      */
273     wait_for_aqueue();
274 }

276     if (closeout_audit(cmd, fcn) == -1)
277         (void) fprintf(stderr, "%s: can't turn off auditd\n",
278             argv[0]);

280     if (cmd == A_SHUTDOWN || cmd == A_REBOOT)
281         (void) creat(SMF_RST, 0777);
282 }

284 (void) adt_free_event(event);
285 if (uadmin(cmd, fcn, mdep) < 0) {
286     perror("uadmin");

288     (void) unlink(SMF_RST);

290     return (1);
291 }

293 /* If returning from a suspend, audit thaw */
294 if ((cmd == A_FREEZE) &&
295     ((fcn == AD_FORCE) ||
296      (fcn == AD_REUSABLE) ||
297      (fcn == AD_SUSPEND_TO_DISK) ||
298      (fcn == AD_SUSPEND_TO_RAM))) {
299     if ((event = adt_alloc_event(ah, ADT_uadmin_thaw)) == NULL) {
300         (void) fprintf(stderr, "%s: can't allocate thaw audit "
301             "event\n", argv[0]);
302     }
303     event->adt_uadmin_thaw.fcn = fcn_id;
304     if (adt_put_event(event, ADT_SUCCESS, 0) != 0) {
305         (void) fprintf(stderr, "%s: can't put thaw audit "
306             "event\n", argv[0]);
307     }
308     (void) adt_free_event(event);
309 }
310 (void) adt_end_session(ah);

312     return (0);
313 }

315 static int
316 closeout_audit(int cmd, int fcn)
317 {
318     if (!adt_audit_state(AUC_AUDITING)) {
319         /* auditd not running, just return */
320         return (0);
321     }
322     switch (cmd) {
323     case A_SHUTDOWN:
324         switch (fcn) {
325         case AD_FASTREBOOT_DRYRUN:

```

```

326         /* No system discontinuity, don't turn off auditd */
327         return (0);
328     default:
329         break; /* For all the other shutdown functions */
330     }
331     /* FALLTHROUGH */
332 case A_REBOOT:
333 case A_DUMP:
334     /* system shutting down, turn off auditd */
335     return (turnoff_auditd());
336 case A_REMOUNT:
337 case A_SWAPCTL:
338 case A_FTRACE:
339 case A_CONFIG:
340     /* No system discontinuity, don't turn off auditd */
341     return (0);
342 case A_FREEZE:
343     switch (fcn) {
344     case AD_CHECK_SUSPEND_TO_DISK: /* AD_CHECK */
345     case AD_CHECK_SUSPEND_TO_RAM:
346     case AD_REUSEINIT:
347     case AD_REUSEFINI:
348         /* No system discontinuity, don't turn off auditd */
349         return (0);
350     case AD_REUSABLE:
351     case AD_SUSPEND_TO_DISK: /* AD_COMPRESS */
352     case AD_SUSPEND_TO_RAM:
353     case AD_FORCE:
354         /* suspend the system, change audit files */
355         return (change_audit_file());
356     default:
357         return (0); /* not an audit error */
358     }
359 default:
360     return (0); /* not an audit error */
361 }
362 }

364 static int
365 turnoff_auditd(void)
366 {
367     int rc;
368     int retries = RETRY_COUNT;

370     if ((rc = (int)fork()) == 0) {
371         (void) execl("/usr/sbin/audit", "audit", "-T", NULL);
372         (void) fprintf(stderr, "error disabling auditd: %s\n",
373             strerror(errno));
374         _exit(-1);
375     } else if (rc == -1) {
376         (void) fprintf(stderr, "error disabling auditd: %s\n",
377             strerror(errno));
378         return (-1);
379     }

381     /*
382      * wait for auditd to finish its work. auditd will change the
383      * auditstart from AUC_AUDITING (auditd up and running) to
384      * AUC_NOAUDIT. Other states are errors, so we're done as well.
385      */
386     do {
387         int auditstate;

389         rc = -1;
390         if ((auditon(A_GETCOND, (caddr_t)&auditstate,
391             sizeof (auditstate)) == 0) &&

```

```
392         (auditstate == AUC_AUDITING)) {
393             retries--;
394             (void) sleep(1);
395         } else {
396             rc = 0;
397         }
398     } while ((rc != 0) && (retries != 0));
399
400     return (rc);
401 }
402
403 static int
404 change_audit_file(void)
405 {
406     pid_t    pid;
407
408     if ((pid = fork()) == 0) {
409         (void) execl("/usr/sbin/audit", "audit", "-n", NULL);
410         (void) fprintf(stderr, "error changing audit files: %s\n",
411             strerror(errno));
412         _exit(-1);
413     } else if (pid == -1) {
414         (void) fprintf(stderr, "error changing audit files: %s\n",
415             strerror(errno));
416         return (-1);
417     } else {
418         pid_t    rc;
419         int      retries = RETRY_COUNT;
420
421         /*
422          * Wait for audit(1M) -n process to complete
423          */
424         do {
425             if ((rc = waitpid(pid, NULL, WNOHANG)) == pid) {
426                 return (0);
427             } else if (rc == -1) {
428                 return (-1);
429             } else {
430                 (void) sleep(1);
431                 retries--;
432             }
433         } while (retries != 0);
434     }
435     return (-1);
436 }
437
438 }
439
440 static void
441 wait_for_aqueue()
442 {
443     au_stat_t    au_stat;
444     int          retries = 10;
445
446     while (retries-- && auditon(A_GETSTAT, (caddr_t)&au_stat, NULL) == 0) {
447         if (au_stat.as_enqueue == au_stat.as_written) {
448             break;
449         }
450         (void) sleep(1);
451     }
452 }
```

new/usr/src/lib/libbsm/audit_event.txt

1

28357 Sat Sep 14 15:00:26 2013
new/usr/src/lib/libbsm/audit_event.txt

XXX Rearchitect and replace interrupt distribution

```
1 #
2 # Copyright (c) 1992, 2010, Oracle and/or its affiliates. All rights reserved.
3 #
4 #
5 # CDDL HEADER START
6 #
7 # The contents of this file are subject to the terms of the
8 # Common Development and Distribution License (the "License").
9 # You may not use this file except in compliance with the License.
10 #
11 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
12 # or http://www.opensolaris.org/os/licensing.
13 # See the License for the specific language governing permissions
14 # and limitations under the License.
15 #
16 # When distributing Covered Code, include this CDDL HEADER in each
17 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
18 # If applicable, add the following below this CDDL HEADER, with the
19 # fields enclosed by brackets "[]" replaced with your own identifying
20 # information: Portions Copyright [yyyy] [name of copyright owner]
21 #
22 # CDDL HEADER END
23 #
24 # Audit Event Database
25 #
26 # File Format:
27 #
28 # event number:event name:event description:event classes (comma separated)
29 #
30 # Used to map audit events to audit classes for preselection and post-selection.
31 # Used by TCB programs that write audit records to preselect audit events
32 # based on event to class mappings.
33 #
34 # NOTE: several events are obsolete but must continue to be defined here for
35 # compatibility reasons. Obsolete events are defined in the "no" (invalid)
36 # class to indicate they will not be generated. Other events in the "no"
37 # class which are not obsolete (but are in this class for other reasons),
38 # are individually noted with a comment for explanation.
39 #
40 # System Administrators: Do NOT modify or add events with an event number less
41 # than 32768. These are reserved by the system.
42 #
43 # 0 Reserved as an invalid event number.
44 # 1 - 2047 Reserved for the Solaris Kernel events.
45 # 2048 - 32767 Reserved for the Solaris TCB programs.
46 # 32768 - 65535 Available for third party TCB applications.
47 #
48 #
49 # Allocation of reserved kernel events:
50 # (NOTE: the kernel event table, and possibly MAX_KEVENTS, must be updated
51 # in audit_events.h when changes are made to kernel events.)
52 # 1 - 511 allocated for Solaris
53 # 512 - 2047 (reserved but not allocated)
54 #
55 # Allocation of user level audit events:
56 # 2048 - 5999 (reserved but not allocated)
57 # 6000 - 9999 allocated for Solaris
58 # 10000 - 32767 (reserved but not allocated)
59 # 32768 - 65535 (Available for third party TCB applications)
60 #
61 #
```

new/usr/src/lib/libbsm/audit_event.txt

2

```
62 # kernel audit events
63 # 1 - 511 allocated for Solaris
64 #
65 0:AUE_NULL:indir system call:no
66 1:AUE_EXIT:exit(2):ps
67 2:AUE_FORKALL:forkall(2):ps
68 # AUE_OPEN is a placeholder and will not be generated
69 3:AUE_OPEN:open(2) - place holder:no
70 4:AUE_CREAT:creat(2):no
71 5:AUE_LINK:link(2):fc
72 6:AUE_UNLINK:unlink(2):fd
73 7:AUE_EXEC:exec(2):no
74 8:AUE_CHDIR:chdir(2):pm
75 9:AUE_MKNOD:mknod(2):fc
76 10:AUE_CHMOD:chmod(2):fm
77 11:AUE_CHOWN:chown(2):fm
78 12:AUE_UMOUNT:umount(2) - old version:as
79 13:AUE_JUNK:junk:no
80 14:AUE_ACCESS:access(2):fa
81 15:AUE_KILL:kill(2):pm
82 16:AUE_STAT:stat(2):fa
83 17:AUE_LSTAT:lstat(2):fa
84 18:AUE_ACCT:acct(2):as
85 19:AUE_MCTL:mctl(2):no
86 20:AUE_REBOOT:reboot(2):no
87 21:AUE_SYMLINK:symlink(2):fc
88 22:AUE_READLINK:readlink(2):fr
89 23:AUE_EXECVE:execve(2):ps,ex
90 24:AUE_CHROOT:chroot(2):pm
91 25:AUE_VFORK:vfork(2):ps
92 26:AUE_SETGROUPS:setgroups(2):pm
93 27:AUE_SETPGRP:setpgrp(2):pm
94 28:AUE_SWAPON:swapon(2):no
95 29:AUE_SETHOSTNAME:sethostname(2):no
96 30:AUE_FCNTL:fcntl(2):fm
97 31:AUE_SETPRIORITY:setpriority(2):no
98 32:AUE_CONNECT:connect(2):nt
99 33:AUE_ACCEPT:accept(2):nt
100 34:AUE_BIND:bind(2):nt
101 35:AUE_SETSOCKOPT:setsockopt(2):nt
102 36:AUE_VTRACE:vtrace(2):no
103 37:AUE_SETTIMEOFDAY:settimeofday(2):no
104 38:AUE_FCHOWN:fchown(2):fm
105 39:AUE_FCHMOD:fchmod(2):fm
106 40:AUE_SETREUID:setreuid(2):pm
107 41:AUE_SETREGID:setregid(2):pm
108 42:AUE_RENAME:rename(2):fc,fd
109 43:AUE_TRUNCATE:truncate(2):no
110 44:AUE_FTRUNCATE:ftruncate(2):no
111 45:AUE_FLOCK:flock(2):no
112 46:AUE_SHUTDOWN:shutdown(2):nt
113 47:AUE_MKDIR:mkdir(2):fc
114 48:AUE_RMDIR:rmdir(2):fd
115 49:AUE_UTIMES:utimes(2):fm
116 50:AUE_ADJTIME:adjtime(2):as
117 51:AUE_SETRLIMIT:setrlimit(2):ua
118 52:AUE_KILLPG:killpg(2):no
119 53:AUE_NFS_SVC:nfs_svc(2):no
120 54:AUE_STATFS:statfs(2):fa
121 55:AUE_FSTATFS:fstatfs(2):fa
122 56:AUE_UNMOUNT:unmount(2):no
123 57:AUE_ASYNC_DAEMON:async_daemon(2):no
124 58:AUE_NFS_GETFH:nfs_getfh(2):no
125 59:AUE_SETDOMAINNAME:setdomainname(2):no
126 60:AUE_QUOTACTL:quotactl(2):no
127 61:AUE_EXPORTFS:exportfs(2):no
```



```

128 62:AUE_MOUNT:mount(2):as
129 # AUE_SEMSYS is a placeholder and will not be generated
130 63:AUE_SEMSYS:semsys(2) - place holder:no
131 # AUE_MSGSYS is a placeholder and will not be generated
132 64:AUE_MSGSYS:msgsys(2) - place holder:no
133 # AUE_SHMSYS is a placeholder and will not be generated
134 65:AUE_SHMSYS:shmsys(2) - place holder:no
135 66:AUE_BSMSYS:bsmsys(2) - place holder:no
136 67:AUE_RFSSYS:rfssys(2) - place holder:no
137 68:AUE_FCHDIR:fchdir(2):pm
138 69:AUE_FCHROOT:fchroot(2):pm
139 70:AUE_VPIXSYS:vpixsys(2) - place holder:no
140 71:AUE_PATHCONF:pathconf(2):fa
141 72:AUE_OPEN_R:open(2) - read:fr
142 73:AUE_OPEN_RC:open(2) - read,creat:fc,fr
143 74:AUE_OPEN_RT:open(2) - read,trunc:fd,fr
144 75:AUE_OPEN_RTC:open(2) - read,creat,trunc:fc,fd,fr
145 76:AUE_OPEN_W:open(2) - write:fw
146 77:AUE_OPEN_WC:open(2) - write,creat:fc,fw
147 78:AUE_OPEN_WT:open(2) - write,trunc:fd,fw
148 79:AUE_OPEN_WTC:open(2) - write,creat,trunc:fc,fd,fw
149 80:AUE_OPEN_RW:open(2) - read,write:fr,fw
150 81:AUE_OPEN_RWC:open(2) - read,write,creat:fc,fw,fr
151 82:AUE_OPEN_RWT:open(2) - read,write,trunc:fd,fr,fw
152 83:AUE_OPEN_RWTC:open(2) - read,write,creat,trunc:fc,fd,fw,fr
153 84:AUE_MSGCTL:msgctl(2) - illegal command:ip
154 85:AUE_MSGCTL_RMID:msgctl(2) - IPC_RMID command:ip
155 86:AUE_MSGCTL_SET:msgctl(2) - IPC_SET command:ip
156 87:AUE_MSGCTL_STAT:msgctl(2) - IPC_STAT command:ip
157 88:AUE_MSGGET:msgget(2):ip
158 89:AUE_MSGRCV:msgrcv(2):ip
159 90:AUE_MSGSND:msgsnd(2):ip
160 91:AUE_SHMCTL:shmctl(2) - illegal command:ip
161 92:AUE_SHMCTL_RMID:shmctl(2) - IPC_RMID command:ip
162 93:AUE_SHMCTL_SET:shmctl(2) - IPC_SET command:ip
163 94:AUE_SHMCTL_STAT:shmctl(2) - IPC_STAT command:ip
164 95:AUE_SHMGET:shmget(2):ip
165 96:AUE_SHMAT:shmat(2):ip
166 97:AUE_SHMDT:shmdt(2):ip
167 98:AUE_SEMCTL:semctl(2) - illegal command:ip
168 99:AUE_SEMCTL_RMID:semctl(2) - IPC_RMID command:ip
169 100:AUE_SEMCTL_SET:semctl(2) - IPC_SET command:ip
170 101:AUE_SEMCTL_STAT:semctl(2) - IPC_STAT command:ip
171 102:AUE_SEMCTL_GETNCNT:semctl(2) - GETNCNT command:ip
172 103:AUE_SEMCTL_GETPID:semctl(2) - GETPID command:ip
173 104:AUE_SEMCTL_GETVAL:semctl(2) - GETVAL command:ip
174 105:AUE_SEMCTL_GETALL:semctl(2) - GETALL command:ip
175 106:AUE_SEMCTL_GETZCNT:semctl(2) - GETZCNT command:ip
176 107:AUE_SEMCTL_SETVAL:semctl(2) - SETVAL command:ip
177 108:AUE_SEMCTL_SETALL:semctl(2) - SETALL command:ip
178 109:AUE_SEMGET:semget(2):ip
179 110:AUE_SEMOP:semop(2):ip
180 111:AUE_CORE:process dumped core:fc
181 112:AUE_CLOSE:close(2):cl
182 113:AUE_SYSTEMBOOT:system booted:na
183 114:AUE_ASYNC_DAEMON_EXIT:async_daemon(2) exited:no
184 115:AUE_NFSSVC_EXIT:nfssvc(2) exited:no
185 116:AUE_PFEEXEC:execve(2) with pfexec enabled:ps,ex,ua,as
186 117:AUE_OPEN_S:open(2) - search:fr
187 118:AUE_OPEN_E:open(2) - exec:fr

189 130:AUE_GETAUID:getauid(2):aa
190 131:AUE_SETAUID:setauid(2):aa
191 132:AUE_GETAUDIT:getaudit(2):aa
192 133:AUE_SETAUDIT:setaudit(2):aa
193 134:AUE_GETUSERAUDIT:getuseraudit(2):no

```

```

194 135:AUE_SETUSERAUDIT:setuseraudit(2):no
195 # AUE_AUDITSVC is a placeholder and will not be generated
196 136:AUE_AUDITSVC:auditsvc(2) - place holder:no
197 # AUE_AUDITON is a placeholder and will not be generated
198 138:AUE_AUDITON:auditon(2) - place holder:no
199 139:AUE_AUDITON_GTERMID:auditon(2) - GETTERMID command:no
200 140:AUE_AUDITON_STERMID:auditon(2) - SETTERMID command:no
201 141:AUE_AUDITON_GPOLICY:auditon(2) - get audit policy flags:aa
202 142:AUE_AUDITON_SPOLICY:auditon(2) - set audit policy flags:as
203 143:AUE_AUDITON_GESTATE:auditon(2) - GESTATE command:no
204 144:AUE_AUDITON_SESTATE:auditon(2) - SESTATE command:no
205 145:AUE_AUDITON_GQCTRL:auditon(2) - get queue control parameters:as
206 146:AUE_AUDITON_SQCTRL:auditon(2) - set queue control parameters:as
207 147:AUE_GETKERNSTATE:getkernstate(2):no
208 148:AUE_SETKERNSTATE:setkernstate(2):no
209 149:AUE_GETPORTAUDIT:getportaudit(2):no
210 150:AUE_AUDITSTAT:auditstat(2):no
211 153:AUE_ENTERPROM:enter prom:na
212 154:AUE_EXITPROM:exit prom:na
213 158:AUE_IOCTL:ioctl(2):io
214 173:AUE_ONESIDE:one-sided session record:no
215 174:AUE_MSGGETL:msggetl(2):no
216 175:AUE_MSGRCVL:msgrcvl(2):no
217 176:AUE_MSGSNL:msgsndl(2):no
218 177:AUE_SEMGETL:semgetl(2):no
219 178:AUE_SHMGETL:shmgetl(2):no
220 183:AUE_SOCKET:socket(2):nt
221 184:AUE_SENDTO:sendto(2):nt
222 # AUE_PIPE is a potentially very high-volume event, use with caution
223 185:AUE_PIPE:pipe(2):no
224 186:AUE_SOCKETPAIR:socketpair(2):no
225 187:AUE_SEND:send(2):no
226 188:AUE_SENDMSG:sendmsg(2):nt
227 189:AUE_RECV:recv(2):no
228 190:AUE_RECVMSG:recvmsg(2):nt
229 191:AUE_RECVFROM:recvfrom(2):nt
230 # AUE_READ is a potentially very high-volume event, use with caution
231 192:AUE_READ:read(2):no
232 193:AUE_GETDENTS:getdents(2):no
233 194:AUE_LSEEK:lseek(2):no
234 # AUE_WRITE is a potentially very high-volume event, use with caution
235 195:AUE_WRITE:write(2):no
236 196:AUE_WRITEV:writev(2):no
237 197:AUE_NFS:nfs server:no
238 198:AUE_READV:readv(2):no
239 199:AUE_OLSTAT:old stat(2):no
240 200:AUE_SETUID:setuid(2):pm
241 201:AUE_STIME:old stime(2):as
242 202:AUE_OLTIME:old utime(2):no
243 203:AUE_NICE:old nice(2):pm
244 204:AUE_OSETPRG:old setprg(2):no
245 205:AUE_SETGID:old setgid(2):pm
246 206:AUE_READL:readl(2):no
247 207:AUE_READVL:readvl(2):no
248 208:AUE_FSTAT:fstat(2):no
249 209:AUE_DUP2:dup2(2):no
250 # AUE_MMAP is a potentially very high-volume event, use with caution
251 210:AUE_MMAP:mmap(2):no
252 # AUE_AUDIT is a potentially very high-volume event, use with caution
253 211:AUE_AUDIT:audit(2):no
254 212:AUE_PRIOCNLSYS:priocnlsys(2):pm
255 213:AUE_MUNMAP:munmap(2):cl
256 214:AUE_SETEGID:setegid(2):pm
257 215:AUE_SETEUID:seteuid(2):pm
258 216:AUE_PUTMSG:putmsg(2):nt
259 217:AUE_GETMSG:getmsg(2):nt

```

```

260 218:AUE_PUTMSG:putmsg(2):nt
261 219:AUE_GETMSG:getmsg(2):nt
262 # AUE_AUDITSYS is a placeholder and will not be generated
263 220:AUE_AUDITSYS:audit system calls place holder:no
264 221:AUE_AUDITON_GETMASK:auditon(2) - get kernel mask:aa
265 222:AUE_AUDITON_SETMASK:auditon(2) - set kernel mask:as
266 223:AUE_AUDITON_GETCWD:auditon(2) - get current working directory:aa,as
267 224:AUE_AUDITON_GETCAR:auditon(2) - get current active root:aa,as
268 225:AUE_AUDITON_GETSTAT:auditon(2) - get audit statistics:as
269 226:AUE_AUDITON_SETSTAT:auditon(2) - reset audit statistics:as
270 227:AUE_AUDITON_SETUMASK:auditon(2) - set mask per audit uid:as
271 228:AUE_AUDITON_SETSMASK:auditon(2) - set mask per session ID:as
272 229:AUE_AUDITON_GETCOND:auditon(2) - get audit state:aa
273 230:AUE_AUDITON_SETCOND:auditon(2) - set audit state:as
274 231:AUE_AUDITON_GETCLASS:auditon(2) - get event class:aa,as
275 232:AUE_AUDITON_SETCCLASS:auditon(2) - set event class:as
276 233:AUE_FUSERS:utssys(2) - fusers:fa
277 234:AUE_STATVFS:statvfs(2):fa
278 235:AUE_XSTAT:xstat(2):no
279 236:AUE_LXSTAT:lxstat(2):no
280 237:AUE_LCHOWN:lchown(2):fm
281 238:AUE_MEMCNTL:memcntl(2):ot
282 239:AUE_SYSINFO:sysinfo(2):as
283 240:AUE_XMKNOD:xmknod(2):no
284 241:AUE_FORK1:fork1(2):ps
285 # AUE_MODCTL is a placeholder and will not be generated
286 242:AUE_MODCTL:modctl(2) system call place holder:no
287 243:AUE_MODLOAD:modctl(2) - load module:as
288 244:AUE_MODUNLOAD:modctl(2) - unload module:as
289 # AUE_MODCONFIG is a place holder and will not be generated
290 245:AUE_MODCONFIG:modctl(2) - no longer generated:no
291 246:AUE_MODADDMJ:modctl(2) - bind module:as
292 247:AUE_SOCKETACCEPT:getmsg-accept:nt
293 248:AUE_SOCKETCONNECT:putmsg-connect:nt
294 249:AUE_SOCKETSEND:putmsg-send:nt
295 250:AUE_SOCKETRECEIVE:getmsg-receive:nt
296 251:AUE_ACLSET:acl(2) - SETACL command:fm
297 252:AUE_FACLSET:facl(2) - SETACL command:fm
298 # AUE_DOORFS is a placeholder and will not be generated
299 253:AUE_DOORFS:doorfs(2) - system call place holder:no
300 254:AUE_DOORFS_DOOR_CALL:doorfs(2) - DOOR_CALL:ip
301 255:AUE_DOORFS_DOOR_RETURN:doorfs(2) - DOOR_RETURN:ip
302 256:AUE_DOORFS_DOOR_CREATE:doorfs(2) - DOOR_CREATE:ip
303 257:AUE_DOORFS_DOOR_REVOKE:doorfs(2) - DOOR_REVOKE:ip
304 258:AUE_DOORFS_DOOR_INFO:doorfs(2) - DOOR_INFO:ip
305 259:AUE_DOORFS_DOOR_CRED:doorfs(2) - DOOR_CRED:ip
306 260:AUE_DOORFS_DOOR_BIND:doorfs(2) - DOOR_BIND:ip
307 261:AUE_DOORFS_DOOR_UNBIND:doorfs(2) - DOOR_UNBIND:ip
308 262:AUE_P_ONLINE:p_online(2):as
309 263:AUE_PROCESSOR_BIND:processor_bind(2):as
310 264:AUE_INST_SYNC:inst_sync(2):as
311 265:AUE_SOCKETCONFIG:configure socket:nt
312 266:AUE_SETAUDIT_ADDR:setaudit_addr(2):aa
313 267:AUE_GETAUDIT_ADDR:getaudit_addr(2):aa
314 268:AUE_UMOUNT2:umount2(2):as
315 # AUE_FSAT and all AUE_OPENAT_* codes are obsolete and will not be generated
316 269:AUE_FSAT:fsat(2) - place holder:no
317 270:AUE_OPENAT_R:openat(2) - read:no
318 271:AUE_OPENAT_RC:openat(2) - read,creat:no
319 272:AUE_OPENAT_RT:openat(2) - read,trunc:no
320 273:AUE_OPENAT_RTC:openat(2) - read,creat,trunc:no
321 274:AUE_OPENAT_W:openat(2) - write:no
322 275:AUE_OPENAT_WC:openat(2) - write,creat:no
323 276:AUE_OPENAT_WT:openat(2) - write,trunc:no
324 277:AUE_OPENAT_WTC:openat(2) - write,creat,trunc:no
325 278:AUE_OPENAT_RW:openat(2) - read,write:no

```

```

326 279:AUE_OPENAT_RWC:openat(2) - read,write,creat:no
327 280:AUE_OPENAT_RWT:openat(2) - read,write,trunc:no
328 281:AUE_OPENAT_RWTC:openat(2) - read,write,creat,trunc:no
329 282:AUE_RENAMEAT:renameat(2):no
330 283:AUE_FSTATAT:fstatat(2):no
331 284:AUE_FCHOWNAT:fchownat(2):no
332 285:AUE_FUTIMESAT:futimesat(2):no
333 286:AUE_UNLINKAT:unlinkat(2):no
334 287:AUE_CLOCK_SETTIME:clock_settime(3RT):as
335 288:AUE_NTP_ADJTIME:ntp_adjtime(2):as
336 289:AUE_SETPPRIV:setppriv(2):pm
337 290:AUE_MODDEVPLCY:modctl(2) - configure device policy:as
338 291:AUE_MODADDPRIV:modctl(2) - configure additional privilege:as
339 292:AUE_CRYPTOADM:kernel cryptographic framework:as
340 293:AUE_CONFIGKSSL:configure kernel SSL:as
341 294:AUE_BRANDSYS:brandsys(2):ot
342 295:AUE_PF_POLICY_ADDRULE:Add IPsec policy rule:as
343 296:AUE_PF_POLICY_DELRULE>Delete IPsec policy rule:as
344 297:AUE_PF_POLICY_CLONE:Clone IPsec policy:as
345 298:AUE_PF_POLICY_FLIP:Flip IPsec policy:as
346 299:AUE_PF_POLICY_FLUSH:Flush IPsec policy rules:as
347 300:AUE_PF_POLICY_ALGS:Update IPsec algorithms:as
348 # AUE_PORTFS is a placeholder and won't be generated.
349 301:AUE_PORTFS:portfs(2) - file events source - place holder:no
350 #
351 302:AUE_LABELSYS_TNRH:tnrh(2) - config TN remote host cache:as
352 303:AUE_LABELSYS_TNRHTP:tnrhttp(2) - config TN remote host template cache:as
353 304:AUE_LABELSYS_TNMLP:tnmlp(2) - config TN multi-level port entry:as
354 #
355 305:AUE_PORTFS_ASSOCIATE:portfs(2) - file events source - PORT_ASSOCIATE:fa
356 306:AUE_PORTFS_DISSOCIATE:portfs(2) - file events source - PORT_DISSOCIATE:fa
357 #
358 307:AUE_SETSID:setsid(2):pm
359 308:AUE_SETPGID:setpgid(2):pm
360 309:AUE_FACCESSAT:faccessat(2):no
361 310:AUE_AUDITON_GETAMASK:auditon(2) - get default user preselection mask:aa
362 311:AUE_AUDITON_SETAMASK:auditon(2) - set default user preselection mask:as
363 #
364 # user level audit events
365 # 2048 - 6143 Reserved
366 #
367 # 6000 - 7999 allocated for Solaris
368 #
369 6144:AUE_at_create:at-create atjob:ua
370 6145:AUE_at_delete:at-delete atjob (at or atrm):ua
371 6146:AUE_at_perm:at-permission:no
372 6147:AUE_cron_invoke:cron-invoke:ua
373 6148:AUE_crontab_create:crontab-crontab created:ua
374 6149:AUE_crontab_delete:crontab-crontab deleted:ua
375 6150:AUE_crontab_perm:crontab-persmission:no
376 6151:AUE_inetd_connect:inetd connect:na
377 6152:AUE_login:login - local:lo
378 6153:AUE_logout:logout:lo
379 6154:AUE_telnet:login - telnet:lo
380 6155:AUE_rlogin:login - rlogin:lo
381 6156:AUE_mountd_mount:mount:na
382 6157:AUE_mountd_umount:umount:na
383 6158:AUE_rshd:rsh access:lo
384 6159:AUE_su:su:lo
385 6160:AUE_halt_solaris:halt(1m):ss
386 6161:AUE_reboot_solaris:reboot(1m):ss
387 6162:AUE_rexecd:rexecd:lo
388 6163:AUE_passwd:passwd:lo
389 6164:AUE_rexd:rexd:lo
390 6165:AUE_ftpd:ftp access:lo
391 6166:AUE_init_solaris:init(1m):ss

```

```
392 6167:AUE_uadmin_solaris:uadmin(1m):no
393 6168:AUE_shutdown_solaris:shutdown(1b):ss
394 6169:AUE_poweroff_solaris:poweroff(1m):ss
395 6170:AUE_crontab_mod:crontab-modify:ua
396 6171:AUE_ftpd_logout:ftp logout:lo
397 6172:AUE_ssh_login:ssh:lo
398 6173:AUE_role_login:role login:lo
399 6180:AUE_prof_cmd:profile command:ua,as
400 6181:AUE_filesystem_add:add filesystem:as
401 6182:AUE_filesystem_delete:delete filesystem:as
402 6183:AUE_filesystem_modify:modify filesystem:as
403 6184:AUE_network_add:add network attributes:as
404 6185:AUE_network_delete:delete network attributes:as
405 6186:AUE_network_modify:modify network attributes:as
406 6187:AUE_printer_add:add printer:as
407 6188:AUE_printer_delete:delete printer:as
408 6189:AUE_printer_modify:modify printer:as
409 6190:AUE_scheduledjob_add:add scheduled job:ua
410 6191:AUE_scheduledjob_delete:delete scheduled job:ua
411 6192:AUE_scheduledjob_modify:modify scheduled job:ua
412 6193:AUE_serialport_add:add serial port:as
413 6194:AUE_serialport_delete:delete serial port:as
414 6195:AUE_serialport_modify:modify serial port:as
415 6196:AUE_usermgr_add:add user/user attributes:ua
416 6197:AUE_usermgr_delete:delete user/user attributes:ua
417 6198:AUE_usermgr_modify:modify user/user attributes:ua
418 6199:AUE_uauth:authorization used:ua,as
419 6200:AUE_allocate_succ:allocate-device success:ot
420 6201:AUE_allocate_fail:allocate-device failure:ot
421 6202:AUE_deallocate_succ:deallocate-device success:ot
422 6203:AUE_deallocate_fail:deallocate-device failure:ot
423 6205:AUE_listdevice_succ:allocate-list devices success:ot
424 6206:AUE_listdevice_fail:allocate-list devices failure:ot
425 6207:AUE_create_user:create user:no
426 6208:AUE_modify_user:modify user:no
427 6209:AUE_delete_user:delete user:no
428 6210:AUE_disable_user:disable user:no
429 6211:AUE_enable_user:enable user:no
430 6212:AUE_newgrp_login:newgrp login:lo
431 6213:AUE_admin_authenticate:admin login:lo
432 6214:AUE_kadmind_auth:authenticated kadmind request:ua
433 6215:AUE_kadmind_unauth:unauthenticated kadmind req:ua
434 6216:AUE_krb5kdc_as_req:kdc authentication svc request:ap
435 6217:AUE_krb5kdc_tgs_req:kdc tkt-grant svc request:ap
436 6218:AUE_krb5kdc_tgs_req_2ndtktm:kdc tgs 2ndtkt mismtch:ap
437 6219:AUE_krb5kdc_tgs_req_alt_tgt:kdc tgs issue alt tgt:ap
438 6220:AUE_smserverd:smsserverd:ot
439 6221:AUE_screenlock:screenlock - lock:lo
440 6222:AUE_screenunlock:screenlock - unlock:lo
441 6223:AUE_zone_state:zoneadmd:ss
442 6224:AUE_inetd_copylimit:inetd copylimit:na
443 6225:AUE_inetd_failrate:inetd failrate:na
444 6226:AUE_inetd_ratelimit:inetd ratelimit:na
445 6227:AUE_zlogin:login - zlogin:lo
446 6228:AUE_su_logout:su logout:lo
447 6229:AUE_role_logout:role logout:lo
448 6230:AUE_attach:attach device:ot
449 6231:AUE_detach:detach device:ot
450 6232:AUE_remove:remove/eject device:ot
451 6233:AUE_pool_import:import device into pool:ot
452 6234:AUE_pool_export:export device from pool:ot
453 6235:AUE_dladm_create_secobj:create network security object:as,cy
454 6236:AUE_dladm_delete_secobj:delete network security object:as,cy
455 6237:AUE_uadmin_shutdown:uadmin(1m) - shutdown:ss
456 6238:AUE_uadmin_reboot:uadmin(1m) - reboot:ss
457 6239:AUE_uadmin_dump:uadmin(1m) - dump:ss
```

```
458 6240:AUE_uadmin_freeze:uadmin(1m) - freeze:ss
459 6241:AUE_uadmin_remount:uadmin(1m) - remount:ss
460 6242:AUE_uadmin_ftrace:uadmin(1m) - ftrace:ss
461 6243:AUE_uadmin_swapctl:uadmin(1m) - swapctl:ss
462 6244:AUE_smbd_session:smbd(1m) session setup:lo
463 6245:AUE_smbd_logoff:smbd(1m) session logoff:lo
464 6246:AUE_vscan_quarantine:vscand(1m) quarantine infected file:na
465 6247:AUE_ndmp_connect:ndmp connect:na
466 6248:AUE_ndmp_disconnect:ndmp disconnect:na
467 6249:AUE_ndmp_backup:ndmp backup:na
468 6250:AUE_ndmp_restore:ndmp restore:na
469 6251:AUE_cpu_ondemand:set ondemand CPU freq governor:ss
470 6252:AUE_cpu_performance:set max CPU freq governor:ss
471 6253:AUE_cpu_threshold:set CPU freq threshold:ss
472 6254:AUE_uadmin_thaw:uadmin(1m) - thaw after freeze:ss,na
473 6255:AUE_uadmin_config:uadmin(1m) - config:ss

475 #
476 # SMF(5) svc.configd events (svcadm(1M) related)
477 #
478 6260:AUE_smf_enable:persistently enable service instance:ss
479 6261:AUE_smf_tmp_enable:temporarily enable service instance:ss
480 6262:AUE_smf_disable:persistently disable service instance:ss
481 6263:AUE_smf_tmp_disable:temporarily disable service instance:ss
482 6264:AUE_smf_restart:restart service instance:ss
483 6265:AUE_smf_refresh:refresh service instance:ss
484 6266:AUE_smf_clear:clear service instance state:ss
485 6267:AUE_smf_degrade:set service instance degraded state:ss
486 6268:AUE_smf_immediate_degrade:immediately set service instance degraded state:ss
487 6269:AUE_smf_maintenance:set service instance persistent maintenance state:ss
488 6270:AUE_smf_immediate_maintenance:immediately set service instance persistent m
489 6271:AUE_smf_immtmp_maintenance:immediately set service instance temporary maint
490 6272:AUE_smf_tmp_maintenance:set service instance maintenance temporary state:ss
491 6273:AUE_smf_milestone:set service management facility milestone:ss
492 #
493 # SMF(5) svc.configd miscellaneous events
494 #
495 6275:AUE_smf_read_prop:read restricted access property value:as
496 #
497 # SMF(5) svc.configd events (svccfg(1M) related)
498 #
499 6280:AUE_smf_create:create service instance object:as
500 6281:AUE_smf_delete:delete service instance object:as
501 6282:AUE_smf_create_pg:create persistent service property group:as
502 6283:AUE_smf_create_npg:create non-persistent service property group:as
503 6284:AUE_smf_delete_pg:delete persistent service property group:as
504 6285:AUE_smf_delete_npg:delete non-persistent service property group:as
505 6286:AUE_smf_create_snap:create repository snapshot:as
506 6287:AUE_smf_delete_snap:delete repository snapshot:as
507 6288:AUE_smf_attach_snap:attach repository snapshot:as
508 6289:AUE_smf_annotation:annotate transaction:as,ss
509 6290:AUE_smf_create_prop:create service instance property:as
510 6291:AUE_smf_change_prop:change service instance property:as
511 6292:AUE_smf_delete_prop:delete service instance property:as
512 #
513 # nwamd(1M) events
514 #
515 6300:AUE_nwam_enable:enable nwam profile object:ss
516 6301:AUE_nwam_disable:disable nwam profile object:ss
517 #
518 # ilbd(1M) events
519 #
520 6310:AUE_ilb_create_healthcheck:create ILB health check:as
521 6311:AUE_ilb_delete_healthcheck:delete ILB health check:as
522 6312:AUE_ilb_create_rule:create ILB rule:as
523 6313:AUE_ilb_delete_rule:delete ILB rule:as
```

```
524 6314:AUE_ilb_disable_rule:disable ILB rule:as
525 6315:AUE_ilb_enable_rule:enable ILB rule:as
526 6316:AUE_ilb_add_server:add ILB server:as
527 6317:AUE_ilb_disable_server:disable ILB server:as
528 6318:AUE_ilb_enable_server:enable ILB server:as
529 6319:AUE_ilb_remove_server:remove ILB server:as
530 6320:AUE_ilb_create_servergroup:create ILB server group:as
531 6321:AUE_ilb_delete_servergroup:delete ILB server group:as
532 #
533 # netcfgd(1M) events
534 #
535 6330:AUE_netcfg_update:create or modify configuration object:ss
536 6331:AUE_netcfg_remove:remove configuration object from repository:ss
537 #
538 # TCSD(8) events
539 #
540 6400:AUE_tpm_takeownership:take ownership of TPM:as
541 6401:AUE_tpm_clearowner:clear ownership of TPM:as
542 6402:AUE_tpm_setoperatorauth:set TPM operator authorization:as
543 6403:AUE_tpm_setownerinstall:set TPM ownership flag:as
544 6404:AUE_tpm_selftestfull:test all TPM protected capabilities:as
545 6405:AUE_tpm_certifyselftest:perform full TPM self-test:as
546 6406:AUE_tpm_continueselftest:complete TPM self-test:as
547 6407:AUE_tpm_ownersetdisable:change the status of TPM disable flag:as
548 6408:AUE_tpm_ownerclear:perform the clear operation under TPM owner auth:as
549 6409:AUE_tpm_disableownerclear:disable TPM_OwnerClear command permanently:as
550 6410:AUE_tpm_forceclear:perform TPM clear operation under physical access:as
551 6411:AUE_tpm_disableforceclear:disable ForceClear execution until next startup:a
552 6412:AUE_tpm_physicaldisable:disable TPM physical presence:as
553 6413:AUE_tpm_physicalenable:enable TPM physical presence:as
554 6414:AUE_tpm_physicaldeactivate:set TPM deactivated flag:as
555 6415:AUE_tpm_settempdeactivated:set volatile TPM deactivated flag to TRUE:as
556 6416:AUE_tpm_settempdeactivated2:set volatile TPM deactivated flag TRUE with aut
557 6417:AUE_tpm_physicalpresence:set the TPM physical presence flag:as
558 6418:AUE_tpm_fieldupgrade:update TPM protected capabilities:as
559 6419:AUE_tpm_resetlockvalue:reset TPM failed authorization attempt lock:as
560 #
561 # hotplugd(1m) events
562 #
563 6500:AUE_hotplug_state:change hotplug connection state:ss
564 6501:AUE_hotplug_set:set hotplug bus private options:ss

566 #
567 # Trusted Extensions events:
568 #
569 9035:AUE_sl_change:Workspace label change:ap
570 9036:AUE_file_relabel:relabel file:fm
571 9037:AUE_file_copy:file copy:fm
572 9038:AUE_file_move:file move:no
573 9039:AUE_sel_mgr_xfer:selection manager transfer:fm
574 9101:AUE_ClientConnect:client connection to x server:lo
575 9102:AUE_ClientDisconnect:client disconn. from x server:lo
576 9120:AUE_ChangeProperty:XChangeProperty(3X11):xc
577 9121:AUE_DeleteProperty:XDeleteProperty(3X11):xc
578 9137:AUE_GrabServer:XGrabServer(3X11):ot
579 9138:AUE_UngrabServer:XUngrabServer(3X11):ot
580 9146:AUE_SetFontPath:XSetFontPath(3X11):ot
581 9173:AUE_InstallColormap:XInstallColormap(3X11):ot
582 9174:AUE_UninstallColormap:XUninstallColormap(3X11):xp
583 9193:AUE_SetScreenSaver:XSetScreenSaver(3X11):xp
584 9194:AUE_ChangeHosts:XChangeHosts(3X11):ot
585 9195:AUE_SetAccessControl:XSetAccessControl(3X11):xp
586 9196:AUE_SetCloseDownMode:XSetCloseDownMode(3X11):xs
587 9197:AUE_KillClient:XKillClient(3X11):xc
588 9202:AUE_XExtensions:X server extensions:xp
589 9103:AUE_CreateWindow:XCreateWindow(3X11):xc
```

```
590 9104:AUE_ChangeWindowAttributes:XChangeWindowAttributes(3X11):xp
591 9105:AUE_GetWindowAttributes:XGetWindowAttributes(3X11):xp
592 9106:AUE_DestroyWindow:XDestroyWindow(3X11):xc
593 9107:AUE_DestroySubwindows:XDestroySubwindows(3X11):xc
594 9108:AUE_ChangeSaveSet:XChangeSaveSet(3X11):xp
595 9109:AUE_ReparentWindow:XReparentWindow(3X11):xp
596 9110:AUE_MapWindow:XMapWindow(3X11):xp
597 9111:AUE_MapSubwindows:XMapSubwindows(3X11):xp
598 9112:AUE_UnmapWindow:XUnmapWindow(3X11):xp
599 9113:AUE_UnmapSubwindows:XUnmapSubwindows(3X11):xp
600 9114:AUE_ConfigureWindow:XConfigureWindow(3X11):xp
601 9115:AUE_CirculateWindow:XCirculateWindow(3X11):xp
602 9116:AUE_GetGeometry:XGetGeometry(3X11):xp
603 9117:AUE_QueryTree:XQueryTree(3X11):xp
604 9118:AUE_InternAtom:XInternAtom(3X11):xs
605 9119:AUE_GetAtomName:XGetAtomName(3X11):xs
606 9122:AUE_GetProperty:XGetProperty(3X11):xp
607 9123:AUE_ListProperties:XListProperties(3X11):xp
608 9124:AUE_SetSelectionOwner:XSetSelectionOwner(3X11):xp
609 9125:AUE_GetSelectionOwner:XGetSelectionOwner(3X11):xs
610 9126:AUE_ConvertSelection:XConvertSelection(3X11):xs
611 9127:AUE_SendEvent:XSendEvent(3X11):xs
612 9128:AUE_GrabPointer:XGrabPointer(3X11):xs
613 9129:AUE_UngrabPointer:XUngrabPointer(3X11):xs
614 9130:AUE_GrabButton:XGrabButton(3X11):xp
615 9131:AUE_UngrabButton:XUngrabButton(3X11):xs
616 9132:AUE_ChangeActivePointerGrab:XChangeActivePointerGrab(3X11):xs
617 9133:AUE_GrabKeyboard:XGrabKeyboard(3X11):xp
618 9134:AUE_UngrabKeyboard:XUngrabKeyboard(3X11):xs
619 9135:AUE_GrabKey:XGrabKey(3X11):xp
620 9136:AUE_UngrabKey:XUngrabKey(3X11):xp
621 9139:AUE_QueryPointer:XQueryPointer(3X11):xp
622 9140:AUE_GetMotionEvents:XGetMotionEvents(3X11):xp
623 9141:AUE_TranslateCoords:XTranslateCoords(3X11):xp
624 9142:AUE_WarpPointer:XWarpPointer(3X11):xs
625 9143:AUE_SetInputFocus:XSetInputFocus(3X11):xs
626 9144:AUE_GetInputFocus:XGetInputFocus(3X11):xs
627 9145:AUE_QueryKeymap:XQueryKeymap(3X11):xp
628 9147:AUE_FreePixmap:XFreePixmap(3X11):xc
629 9148:AUE_ChangeGC:XChangeGC(3X11):xp
630 9149:AUE_CopyGC:XCopyGC(3X11):xp
631 9150:AUE_SetDashes:XSetDashes(3X11):xp
632 9151:AUE_SetClipRectangles:XSetClipRectangles(3X11):xp
633 9152:AUE_FreeGC:XFreeGC(3X11):xc
634 9153:AUE_ClearArea:XClearArea(3X11):xp
635 9154:AUE_CopyArea:XCopyArea(3X11):xs
636 9155:AUE_CopyPlane:XCopyPlane(3X11):xs
637 9156:AUE_PolyPoint:XPolyPoint(3X11):xp
638 9157:AUE_PolyLine:XPolyLine(3X11):xp
639 9158:AUE_PolySegment:XPolySegment(3X11):xp
640 9159:AUE_PolyRectangle:XPolyRectangle(3X11):xs
641 9160:AUE_PolyArc:XPolyArc(3X11):xp
642 9161:AUE_FillPolygon:XFillPolygon(3X11):xp
643 9162:AUE_PolyFillRectangle:XPolyFillRectangle(3X11):xp
644 9163:AUE_PolyFillArc:XPolyFillArc(3X11):xp
645 9164:AUE_PutImage:XPutImage(3X11):xp
646 9165:AUE_GetImage:XGetImage(3X11):xs
647 9166:AUE_PolyText8:XPolyText8(3X11):xp
648 9167:AUE_PolyText16:XPolyText16(3X11):xp
649 9168:AUE_ImageText8:XImageText8(3X11):xp
650 9169:AUE_ImageText16:XImageText16(3X11):xp
651 9170:AUE_CreateColormap:XCreateColormap(3X11):xc
652 9171:AUE_FreeColormap:XFreeColormap(3X11):xc
653 9172:AUE_CopyColormapAndFree:XCopyColormapAndFree(3X11):xp
654 9175:AUE_ListInstalledColormaps:XListInstalledColormaps(3X11):xs
655 9176:AUE_AllocColor:XAllocColor(3X11):xc
```

```
656 9177:AUE_AllocNamedColor:XAllocNamedColor(3X11):xc
657 9178:AUE_AllocColorCells:XAllocColorCells(3X11):xc
658 9179:AUE_AllocColorPlanes:XAllocColorPlanes(3X11):xc
659 9180:AUE_FreeColors:XFreeColors(3X11):xc
660 9181:AUE_StoreColors:XStoreColors(3X11):xp
661 9182:AUE_StoreNamedColor:XStoreNamedColor(3X11):xp
662 9183:AUE_QueryColors:XQueryColors(3X11):xp
663 9184:AUE_LookupColor:XLookupColor(3X11):xp
664 9185:AUE_CreateCursor:XCreateCursor(3X11):xc
665 9186:AUE_CreateGlyphCursor:XCreateGlyphCursor(3X11):xc
666 9187:AUE_FreeCursor:XFreeCursor(3X11):xc
667 9188:AUE_RecolorCursor:XRecolorCursor(3X11):xp
668 9189:AUE_ChangeKeyboardMapping:XChangeKeyboardMapping(3X11):xs
669 9190:AUE_ChangeKeyboardControl:XChangeKeyboardControl(3X11):xs
670 9191:AUE_Bell:XBell(3X11):xs
671 9192:AUE_ChangePointerControl:XChangePointerControl(3X11):xs
672 9198:AUE_RotateProperties:XRotateProperties(3X11):xp
673 9199:AUE_ForceScreenSaver:XForceScreenSaver(3X11):xp
674 9200:AUE_SetPointerMapping:XSetPointerMapping(3X11):xs
675 9201:AUE_SetModifierMapping:XSetModifierMapping(3X11):xs
676 9202:AUE_uadmin_intrd:uadmin(1m) - intrd:ss
677 #endif /* ! codereview */
```

```

*****
95183 Sat Sep 14 15:00:27 2013
new/usr/src/lib/libbsm/common/adt.xml
XXX Rearchitect and replace interrupt distribution
*****
1 <?xml version="1.0" standalone="yes"?>
2 <!DOCTYPE specification SYSTEM "audit.dtd">
3 <!--
4 CDDL HEADER START

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

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

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

21 CDDL HEADER END

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

25 -->

27 <specification>

29 <!-- comments are displayed to stderr if debug is on -->
30 <debug set="off"/>

32 <!-- The order of events is arbitrary EXCEPT generic events must
33 precede their instances -->
34 <!-- The order of entries within an event determine the order
35 data is defined in the external API -->
36 <!-- The order of internal / external is arbitrary -->

38 <!--
39 tags:
40
41 The following top level tags are defined:
<event> <token> <msg_list> <debug>

43 event defines an audit record
44 - id is the record id from audit_uevents.h
45 - reorder="yes" or "no". (default is "no").
46 if "yes" then the order of the tokens to be
47 output does not match the order of the data
48 input. (see order attribute of <entry>)
49 - header defines the header file to contain the external
50 definitions for this event type. The header file
51 name is adt_event_N.h, where N is the value supplied
52 header="0" is for "stable" events, > 0 for new ones.
53 with this attribute. (header="1").
54 - idNo is the number associated with the external
55 name of this event. (For AUE_login, ADT_login is
56 the external name and idNo is the value for
57 ADT_login.)
58 - omit is by default 'no' (i.e., don't omit) and can be
59 'always' or 'JNI'. In the latter case, C interface
60 code is generated but neither Java nor JNI code is.
61 - included text is just a comment

```

```

62 Within an event block, the following tags are defined:
63 <entry>, <debug>, <altname>, <title>, <program>, <see>

65 altname defines the internal name of an audit record; if
66 omitted, the internal name is the same as the
67 external name.

69 title, these tags are used by auditrecord(1M) build to create
70 audit_record_attr database from adt events.
71 see Following example demonstrates their semantics:

73 auditrecord -p passwd
74 passwd <- <title>
75 program various See passwd(1)
76 ^- <program> ^- <see>
77 event ID 6163 AUE_passwd
78 class lo (0x00001000)
79 header
80 subject
81 [text] username...
82 ^- <comment>
83 return

85 If the length of string in any of the given
86 elements is longer than defined, the string is
87 silently truncated to the defined length in the
88 auditrecord(1M) runtime:
89
90 element <= max (non-truncated) string length
91 title <= 46
92 program <= 20
93 see <= 39
94 comment <= unlimited
95 token <= 28

97 entry defines the correspondence between the data
98 supplied by the caller and the token to be
99 output.
100 - id is the data name that shows up in the structures
101 of adt_event.h. If it is a comma separated list,
102 it is the list of names of data to be associated
103 with one output token. (See <external>, below)
104 Within an entry block, the following tags are defined:
105 <internal>, <external>, <debug>, <comment>

107 internal defines the token to be generated.
108 - token is a name that must also be defined with
109 a <token> tag elsewhere in this file. (order is
110 not important).
111 - order="some number" determines the order of the
112 tokens to be output, starting with 1. The subject
113 token is normally order="1". The use is to insure
114 that the order of fields listed in adt_event.h does
115 not change when we arbitrarily change the order of
116 tokens. If the <event reorder="yes"> is not set,
117 order is ignored.
118 - format is a printf-like string that will be used
119 in to format the data supplied by the user.

121 external defines the data to be supplied for creating the
122 token defined via <internal>
123 - opt is one of four values: "required", "optional",
124 "obsolete", or "none". The first two values
125 indicate that this token's data must or may
126 be supplied by the user; the third value is
127 equivalent to "optional" but shows in the

```

```

128      comment that this field is no longer used;
129      the forth value indicates that this token
130      does not require any user-supplied data. If
131      data is required, then a token is always
132      output, while optional data is output only
133      if data is supplied.
134      - type describes the C data type to be associated
135      with the <entry id="dataName">. The following
136      data types are representative:

138          au_asid_t (uint32_t)
139          char
140          char * (blank is optional)
141          char ** (blank is optional)
142          fd_t (int, a file descriptor)
143          uint_t, int, int32_t, uid_t, gid_t
144          uid_t *, gid_t *
145          long, ulong_t
146          m_label_t *
147          pid_t
148          priv_set_t *
149          uint16_t, uint32_t, uint64_t
150          uint32_t *, uint32_t[], uint64_t *
151          msg (not a C type, see below)

153      Below is what Tony said. Above seems to be
154      what is implemented
155          char
156          char * (blank is optional)
157          char ** (blank is optional)
158          int, uid_t, gid_t
159          int *, uid_t *, gid_t *
160          msg (not a C type, see below)
161          time_t
162          uint, uint *

164      The msg type refers to an enumerated type
165      that must be defined via a <msg> description
166      else where in this file. The syntax is
167      special. Example: <external opt="optional"
168      type="msg login_text"/> "login_text" is the
169      id of a <msg_list> descriptor given
170      elsewhere in this file.

172      If the <entry> id is a list, the type must also
173      be a comma-separated list, where the types are
174      in the same order as the id's.
175      If the type is an array, its length must be given
176      explicitly.

178      comment      Used by auditrecord(1M) build to generate
179                   audit_record_attr. Comment is explanation note
180                   printed with token type. Colon (':') may not be
181                   used in a comment. See example above for other
182                   tags related to auditrecord(1M).

184      token      Define allowed token names.
185                 - id is the name of token; this name is used
186                 as an <internal> id.
187                 - a token id name may not end in digits.
188                 - usage is an optional value. At present, only
189                   "TSOL" is defined; it means that this data is
190                   to be used only in Trusted Solaris implementations.
191                   See also example above for "token" tag relation to
192                   the auditrecord(1M) output.

```

```

194      msg_list      Define a set of text strings.
195                   - id is the name to be used for this group of text
196                     strings in adt_event.h
197                   - header is as defined for <event>
198                   - start is a number where produced enum type begins;
199                     ensure msg lists do not overlap
200                   Within a msg_list block, <msg> and <debug> are defined.
201                   The order of <msg> tags in a msg_list is reflected
202                   directly in adt_event.h. Also add ADT_LIST_<id> to
203                   enum adt_msg_list in adt_xlate.h.

205      msg           Define one string.
206                   - id is the name to be used in the enum describing
207                     this set of strings. Convention: use upper case.
208                   The content (text between <msg> and </msg>) is the
209                   actual string. Extra white space, including line
210                   feeds, is ignored. If empty, no output token
211                   is generated unless the <external> opt attribute is
212                   set to "required", in which case a blank text token
213                   is generated.
214                   Within a msg block, <debug> is defined, but has not been
215                   tested and may have no effect.

217      debug         This turns on/off debug messages during the processing
218                   of the xml data. It affects the block within which it
219                   is defined.
220                   - set may have one of two values: "on" or "off". If
221                     set is omitted, the debug state for the current block
222                     is toggled.
223                   The use of the <debug> tag does not affect the output
224                   of data to the various files created, but does generate
225                   potentially large amounts of output to stderr.

227      -->
228      <!--          template for an event record definition

230      <event id="" header="0" idNo="">
231      <entry id="subject">
232      <internal token="subject"/>
233      <external opt="none"/>
234      </entry>
235      <entry id="">
236      <internal token="">
237      <external opt="" type="" />
238      </entry>
239      <entry id="return">
240      <internal token="return"/>
241      <external opt="none"/>
242      </entry>
243      </event>

245      Generic events must precede Instance events; within each
246      group, please group the AUE_* by area and event idNo-s in order,
247      gaps in idNo-s are OK.
248      N.B. Renumbering idNo-s requires recompilation of consumers. See
249      the contracts for whom to notify if/when this happens.
250      -->

252      <!-- generic events -->

254      <!--
255      'omit="always"' means that this record type is not reflected
256      in the generated header and table files.
257      -->

259      <event id="AUE_generic_basic" type="generic" omit="always">

```

```

260      <!--
262      This is a template for the event types that have no tokens
263      other than the header and return. There is no allowed_type
264      list because the template is not externally visible due to the
265      omit="always".
267      -->
268      <entry id="subject">
269        <internal token="subject"/>
270        <external opt="none"/>
271      </entry>
272      <entry id="return">
273        <internal token="return"/>
274        <external opt="none"/>
275      </entry>
276    </event>
278    <event id="AUE_generic_login" type="generic" omit="always">
279      <!--
281      This is a template for the various login event types
282      AUE_login, AUE_ftp, etc which match this template. There is
283      no allowed_type list because the template is not externally
284      visible due to the omit="always".
286      -->
287      <entry id="subject">
288        <internal token="subject"/>
289        <external opt="none"/>
290      </entry>
292      <!-- This field is still in use for SMC until it is cleaned up,
293             it must remain, see login_text msg list at the end of the
294             file.
295      -->
296      <entry id="message">
297        <internal token="text"/>
298        <external opt="optional" type="msg login_text"/>
299        <comment>error message</comment>
300      </entry>
301      <entry id="return">
302        <internal token="return"/>
303        <external opt="none"/>
304      </entry>
305    </event>
307    <!-- generic SMC events -->
309    <event id="AUE_generic_SMC_add" type="generic" omit="always">
310      <entry id="subject">
311        <internal token="subject"/>
312        <external opt="none"/>
313      </entry>
314      <entry id="object_name">
315        <internal token="text"/>
316        <external opt="required" type="char *"/>
317        <comment>object name</comment>
318      </entry>
319      <entry id="domain">
320        <internal token="text"/>
321        <external opt="optional" type="char *"/>
322        <comment>domain</comment>
323      </entry>
324      <entry id="name_service">
325        <internal token="text"/>

```

```

326        <external opt="required" type="char *"/>
327        <comment>name_service</comment>
328      </entry>
329      <entry id="auth_used">
330        <internal token="uauth"/>
331        <external opt="optional" type="char *"/>
332        <comment>authorization used</comment>
333      </entry>
334      <!--
335      This should really be its own token type, not "text"
336      -->
337      <entry id="initial_values">
338        <internal token="text"/>
339        <external opt="required" type="char *"/>
340        <comment>initial values</comment>
341      </entry>
342      <entry id="return">
343        <internal token="return"/>
344        <external opt="none"/>
345      </entry>
346    </event>
348    <event id="AUE_generic_SMC_delete" type="generic" omit="always">
349      <entry id="subject">
350        <internal token="subject"/>
351        <external opt="none"/>
352      </entry>
353      <entry id="object_name">
354        <internal token="text"/>
355        <external opt="required" type="char *"/>
356        <comment>object name</comment>
357      </entry>
358      <entry id="domain">
359        <internal token="text"/>
360        <external opt="optional" type="char *"/>
361        <comment>domain</comment>
362      </entry>
363      <entry id="name_service">
364        <internal token="text"/>
365        <external opt="required" type="char *"/>
366        <comment>name_service</comment>
367      </entry>
368      <entry id="auth_used">
369        <internal token="uauth"/>
370        <external opt="optional" type="char *"/>
371        <comment>authorization used</comment>
372      </entry>
373      <entry id="delete_values">
374        <internal token="text"/>
375        <external opt="required" type="char *"/>
376        <comment>deleted values</comment>
377      </entry>
378      <entry id="return">
379        <internal token="return"/>
380        <external opt="none"/>
381      </entry>
382    </event>
384    <event id="AUE_generic_SMC_modify" type="generic" omit="always">
385      <entry id="subject">
386        <internal token="subject"/>
387        <external opt="none"/>
388      </entry>
389      <entry id="object_name">
390        <internal token="text"/>
391        <external opt="required" type="char *"/>

```



```

392     <comment>object name</comment>
393 </entry>
394 <entry id="domain">
395     <internal token="text"/>
396     <external opt="optional" type="char *"/>
397     <comment>domain</comment>
398 </entry>
399 <entry id="name_service">
400     <internal token="text"/>
401     <external opt="required" type="char *"/>
402     <comment>name_service</comment>
403 </entry>
404 <entry id="auth_used">
405     <internal token="uauth"/>
406     <external opt="optional" type="char *"/>
407     <comment>authorization used</comment>
408 </entry>
409 <entry id="changed_values">
410     <internal token="text"/>
411     <external opt="required" type="char *"/>
412     <comment>changed values</comment>
413 </entry>
414 <entry id="return">
415     <internal token="return"/>
416     <external opt="none"/>
417 </entry>
418 </event>

420 <!-- instances -->

422 <!--
423     Java needed for SMC events. Since the SMC events grow less
424     often than the C related events. They come first. It
425     would be nice to reorder the idNo-s, but that's an ABI
426     change and should rev libbsm version no. If reordered
427     start with 1 and eliminate the comment at the end about
428     the highest idNo.
429 -->
430 <event id="AUE_admin_authenticate" instance_of="AUE_generic_login"
431     header="0" idNo="3">
432     <title>Admin Server Authentication</title>
433     <program>admin (various)</program>
434     <see>SMC, WBEM, or AdminSuite</see>
435 </event>

437 <event id="AUE_filesystem_add" instance_of="AUE_generic_SMC_add"
438     header="0" idNo="4">
439     <title>SMC: filesystem add</title>
440     <program>SMC server</program>
441 </event>
442 <event id="AUE_filesystem_delete" instance_of="AUE_generic_SMC_delete"
443     header="0" idNo="5">
444     <title>SMC: filesystem delete</title>
445     <program>SMC server</program>
446 </event>
447 <event id="AUE_filesystem_modify" instance_of="AUE_generic_SMC_modify"
448     header="0" idNo="6">
449     <title>SMC: filesystem modify</title>
450     <program>SMC server</program>
451 </event>

453 <event id="AUE_network_add" instance_of="AUE_generic_SMC_add"
454     header="0" idNo="7">
455     <title>SMC: network add</title>
456     <program>SMC server</program>
457 </event>

```

```

458 <event id="AUE_network_delete" instance_of="AUE_generic_SMC_delete"
459     header="0" idNo="8">
460     <title>SMC: network delete</title>
461     <program>SMC server</program>
462 </event>
463 <event id="AUE_network_modify" instance_of="AUE_generic_SMC_modify"
464     header="0" idNo="9">
465     <title>SMC: network modify</title>
466     <program>SMC server</program>
467 </event>

469 <event id="AUE_printer_add" instance_of="AUE_generic_SMC_add"
470     header="0" idNo="10">
471     <title>SMC: printer add</title>
472     <program>SMC server</program>
473 </event>
474 <event id="AUE_printer_delete" instance_of="AUE_generic_SMC_delete"
475     header="0" idNo="11">
476     <title>SMC: printer delete</title>
477     <program>SMC server</program>
478 </event>
479 <event id="AUE_printer_modify" instance_of="AUE_generic_SMC_modify"
480     header="0" idNo="12">
481     <title>SMC: printer modify</title>
482     <program>SMC server</program>
483 </event>

485 <!--
486     This is SMC; it's also used in su and should probably be used in
487     desktop role login. If we fix the SMC to not record NO_MSG here,
488     we can fix to record failed user. See su.c and AUE_su.
489 -->
490 <event id="AUE_role_login" instance_of="AUE_generic_login"
491     header="0" idNo="13">
492     <title>RBAC: role login</title>
493     <program>SMC server</program>
494     <program>/usr/bin/su</program>
495 </event>

497 <event id="AUE_scheduledjob_add" instance_of="AUE_generic_SMC_add"
498     header="0" idNo="14">
499     <title>SMC: scheduled job add</title>
500     <program>SMC server</program>
501 </event>
502 <event id="AUE_scheduledjob_delete" instance_of="AUE_generic_SMC_delete"
503     header="0" idNo="15">
504     <title>SMC: scheduled job delete</title>
505     <program>SMC server</program>
506 </event>
507 <event id="AUE_scheduledjob_modify" instance_of="AUE_generic_SMC_modify"
508     header="0" idNo="16">
509     <title>SMC: scheduled job modify</title>
510     <program>SMC server</program>
511 </event>

513 <event id="AUE_serialport_add" instance_of="AUE_generic_SMC_add"
514     header="0" idNo="17">
515     <title>SMC: serial port add</title>
516     <program>SMC server</program>
517 </event>
518 <event id="AUE_serialport_delete" instance_of="AUE_generic_SMC_delete"
519     header="0" idNo="18">
520     <title>SMC: serial port delete</title>
521     <program>SMC server</program>
522 </event>
523 <event id="AUE_serialport_modify" instance_of="AUE_generic_SMC_modify"

```

```

524         header="0" idNo="19">
525         <title>SMC: serial port modify</title>
526         <program>SMC server</program>
527     </event>

529 <!-- This is SMC; should this also be used elsewhere? -->
530     <event id="AUE_uauth" header="0" idNo="20">
531         <title>SMC: Use of Authorization</title>
532         <program>SMC server</program>
533         <entry id="subject">
534             <internal token="subject"/>
535             <external opt="none"/>
536         </entry>
537         <entry id="auth_used">
538             <internal token="uauth"/>
539             <external opt="required" type="char *"/>
540             <comment>authorization used</comment>
541         </entry>
542         <entry id="objectname">
543             <internal token="text"/>
544             <external opt="required" type="char *"/>
545             <comment>object name</comment>
546         </entry>
547         <entry id="return">
548             <internal token="return"/>
549             <external opt="none"/>
550         </entry>
551     </event>

553     <event id="AUE_usermgr_add" instance_of="AUE_generic_SMC_add"
554         header="0" idNo="21">
555         <title>SMC: User Manager add</title>
556         <program>SMC server</program>
557     </event>
558     <event id="AUE_usermgr_delete" instance_of="AUE_generic_SMC_delete"
559         header="0" idNo="22">
560         <title>SMC: User Manager delete</title>
561         <program>SMC server</program>
562     </event>
563     <event id="AUE_usermgr_modify" instance_of="AUE_generic_SMC_modify"
564         header="0" idNo="23">
565         <title>SMC: User Manager modify</title>
566         <program>SMC server</program>
567     </event>
568 <!-- end of Java needed for SMC events -->
569 <!--
570     while not used by SMC logout is used by Lockhart
571 -->
572     <event id="AUE_logout" header="0" idNo="1">
573         <title>login: logout</title>
574         <program>various</program>
575         <see>login(1)</see>
576         <entry id="subject">
577             <internal token="subject"/>
578             <external opt="none"/>
579         </entry>
580 <!--
581     not used by C code, used by Lockhart,
582     get them to change and remove
583     event.user_name("logout " + session.getUserName());
584     from /ws/lockhart-nv-gate/src/bundled/app/webmgmt/lib/services/
585     com/sun/management/services/audit/SolarisAuditEvent_Logout.java
586 -->
587         <entry id="user_name">
588             <internal token="text" format="logout %s"/>
589             <external opt="optional" type="char *"/>

```

```

590         <comment>"logout" username</comment>
591     </entry>
592     <entry id="return">
593         <internal token="return"/>
594         <external opt="none"/>
595     </entry>
596 </event>

599 <!-- C Only events -->
600     <event id="AUE_init_solaris" header="0" idNo="32" omit="JNI">
601         <title>init</title>
602         <program>/sbin/init</program>
603         <program>/usr/sbin/init</program>
604         <program>/usr/sbin/shutdown</program>
605         <entry id="subject">
606             <internal token="subject"/>
607             <external opt="none"/>
608         </entry>
609         <entry id="info">
610             <internal token="text"/>
611             <external opt="optional" type="char *"/>
612             <comment>init level or zone name</comment>
613         </entry>
614         <entry id="return">
615             <internal token="return"/>
616             <external opt="none"/>
617         </entry>
618     </event>

620     <event id="AUE_login" instance_of="AUE_generic_login" header="0"
621         idNo="25" omit="JNI">
622         <title>terminal login</title>
623         <program>/usr/sbin/login</program>
624         <program>/usr/dt/bin/dtlogin</program>
625         <see>login(1)</see>
626         <see>dtlogin</see>
627     </event>
628     <event id="AUE_rlogin" instance_of="AUE_generic_login" header="0"
629         idNo="28" omit="JNI">
630         <title>rlogin</title>
631         <program>/usr/sbin/login</program>
632         <see>login(1) - rlogin</see>
633     </event>
634     <event id="AUE_telnet" instance_of="AUE_generic_login" header="0"
635         idNo="29" omit="JNI">
636         <title>telnet login</title>
637         <program>/usr/sbin/login</program>
638         <see>login(1) - telnet</see>
639     </event>
640     <event id="AUE_ssh" instance_of="AUE_generic_login" header="0"
641         idNo="2" omit="JNI">
642         <program>/usr/lib/ssh/sshd</program>
643     </event>

645     <event id="AUE_zlogin" header="0" idNo="38" omit="JNI">
646         <title>zone login</title>
647         <program>/usr/sbin/login</program>
648         <see>zlogin(1)</see>
649         <entry id="subject">
650             <internal token="subject"/>
651             <external opt="none"/>
652         </entry>
653         <entry id="message">
654             <internal token="text"/>
655             <external opt="optional" type="char *"/>

```

```

656         <comment>error message</comment>
657     </entry>
658     <entry id="return">
659         <internal token="return"/>
660         <external opt="none"/>
661     </entry>
662 </event>

664 <event id="AUE_su" header="0" idNo="30" omit="JNI">
665     <title>su</title>
666     <program>/usr/bin/su</program>
667     <see>su(1M)</see>
668     <entry id="subject">
669         <internal token="subject"/>
670         <external opt="none"/>
671     </entry>
672 <!--
673     should be changed to "fail_user" and su.c updated
674     However, the jni stuff is broken, so for now it's "message"
675 -->
676     <entry id="message">
677         <internal token="text"/>
678         <external opt="optional" type="char *"/>
679         <comment>"user name" of failed new user/role</comment>
680     </entry>
681     <entry id="return">
682         <internal token="return"/>
683         <external opt="none"/>
684     </entry>
685 </event>

687 <event id="AUE_passwd" header="0" idNo="27" omit="JNI">
688     <title>passwd</title>
689     <program>various</program>
690     <see>passwd(1)</see>
691     <entry id="subject">
692         <internal token="subject"/>
693         <external opt="none"/>
694     </entry>
695     <entry id="uid,username">
696         <internal token="user"/>
697         <external opt="optional" type="uid_t,char *"/>
698         <comment>user if different than caller</comment>
699     </entry>
700     <entry id="return">
701         <internal token="return"/>
702         <external opt="none"/>
703     </entry>
704 </event>

706 <event id="AUE_screenlock" instance_of="AUE_generic_basic" header="0"
707     idNo="26" omit="JNI">
708     <program>desktop screen lock</program>
709 </event>
710 <event id="AUE_screenunlock" instance_of="AUE_generic_basic" header="0"
711     idNo="31" omit="JNI">
712     <program>desktop screen unlock</program>
713 </event>

715 <!--
716     AUE_prof_cmd is not supportable for Java due to the structure of
717     the priv token. When and if a Java program needs to generate
718     a priv token, we'll need to look at the data format in the
719     Java code and provide an appropriate java and jni implementation.
720 -->

```

```

722 <event id="AUE_prof_cmd" header="0" idNo="24" omit="JNI">
723     <title>pfexec</title>
724     <program>/usr/bin/pfexec</program>
725     <see>pfexec(1)</see>
726     <entry id="subject">
727         <internal token="subject"/>
728         <external opt="none"/>
729     </entry>
730     <entry id="cwdpath">
731         <internal token="path"/>
732         <external opt="required" type="char *"/>
733         <comment>working directory</comment>
734     </entry>
735     <entry id="cmdpath">
736         <internal token="path"/>
737         <external opt="required" type="char *"/>
738         <comment>command pathname</comment>
739     </entry>
740     <entry id="argc,argv,envp">
741         <internal token="command"/>
742         <external opt="required" type="int,char**,char**"/>
743     </entry>
744     <entry id="proc_auid,proc_euid,proc_egid,proc_ruid,proc_rgid,proc_pid,pr
745         <internal token="process"/>
746         <external opt="required"
747             type="uid_t,uid_t,gid_t,uid_t,gid_t,pid_t,au_asid_t,termid*"/>
748     </entry>
749     <entry id="limit_set">
750         <internal token="priv_limit"/>
751         <external opt="optional" type="priv_set_t*"/>
752     </entry>
753     <entry id="inherit_set">
754         <internal token="priv_inherit"/>
755         <external opt="optional" type="priv_set_t*"/>
756     </entry>
757     <entry id="return">
758         <internal token="return"/>
759         <external opt="none"/>
760     </entry>
761 </event>

763 <event id="AUE_inetd_connect" header="0" idNo="34" omit="JNI">
764     <title>inetd</title>
765     <program>/usr/sbin/inetd</program>
766     <entry id="subject">
767         <internal token="subject"/>
768         <external opt="none"/>
769     </entry>
770     <entry id="service_name">
771         <internal token="text"/>
772         <external opt="optional" type="char *"/>
773         <comment>service name</comment>
774     </entry>
775     <entry id="ip_type,ip_remote_port,ip_local_port,ip_adr">
776         <internal token="tid"/>
777         <external opt="required"
778             type="uint32_t,uint16_t,uint16_t,uint32_t[4]"/>
779         <comment>client address</comment>
780     </entry>
781     <entry id="cmd">
782         <internal token="command_alt"/>
783         <external opt="required" type="char *"/>
784         <comment>inetd command</comment>
785     </entry>
786     <entry id="privileges">
787         <internal token="priv_effective"/>

```

```

788     <external opt="required" type="priv_set_t *"/>
789   </entry>
790   <entry id="return">
791     <internal token="return"/>
792     <external opt="none"/>
793   </entry>
794 </event>

796 <event id="AUE_inetd_ratelimit" header="0" idNo="35" omit="JNI">
797   <title>inetd</title>
798   <program>/usr/sbin/inetd</program>
799   <entry id="subject">
800     <internal token="subject"/>
801     <external opt="none"/>
802   </entry>
803   <entry id="service_name">
804     <internal token="text"/>
805     <external opt="optional" type="char *"/>
806     <comment>service name</comment>
807   </entry>
808   <entry id="limit">
809     <internal token="text"/>
810     <external opt="required" type="char *"/>
811     <comment>limit value</comment>
812   </entry>
813   <entry id="return">
814     <internal token="return"/>
815     <external opt="none"/>
816   </entry>
817 </event>

819 <event id="AUE_inetd_copylimit" header="0" idNo="36" omit="JNI">
820   <title>inetd</title>
821   <program>/usr/sbin/inetd</program>
822   <entry id="subject">
823     <internal token="subject"/>
824     <external opt="none"/>
825   </entry>
826   <entry id="service_name">
827     <internal token="text"/>
828     <external opt="optional" type="char *"/>
829     <comment>service name</comment>
830   </entry>
831   <entry id="limit">
832     <internal token="text"/>
833     <external opt="required" type="char *"/>
834     <comment>limit value</comment>
835   </entry>
836   <entry id="return">
837     <internal token="return"/>
838     <external opt="none"/>
839   </entry>
840 </event>

842 <event id="AUE_inetd_failrate" header="0" idNo="37" omit="JNI">
843   <title>inetd</title>
844   <program>/usr/sbin/inetd</program>
845   <entry id="subject">
846     <internal token="subject"/>
847     <external opt="none"/>
848   </entry>
849   <entry id="service_name">
850     <internal token="text"/>
851     <external opt="optional" type="char *"/>
852     <comment>service name</comment>
853   </entry>

```

```

854   <entry id="values">
855     <internal token="text"/>
856     <external opt="required" type="char *"/>
857     <comment>limit value, interval</comment>
858   </entry>
859   <entry id="return">
860     <internal token="return"/>
861     <external opt="none"/>
862   </entry>
863 </event>

865 <event id="AUE_zone_state" header="0" idNo="33" omit="JNI">
866   <entry id="subject">
867     <internal token="subject"/>
868     <external opt="none"/>
869   </entry>
870   <entry id="new_state">
871     <internal token="text"/>
872     <external opt="required" type="char *"/>
873     <comment>New zone state</comment>
874   </entry>
875   <entry id="zonename">
876     <internal token="zonename"/>
877     <external opt="required" type="char *"/>
878     <comment>zone name</comment>
879   </entry>
880   <entry id="return">
881     <internal token="return"/>
882     <external opt="none"/>
883   </entry>
884 </event>

886 <event id="AUE_su_logout" instance_of="AUE_generic_basic"
887   header="0" idNo="39" omit="JNI">
888   <title>su</title>
889   <program>/usr/bin/su</program>
890   <see>su(1M)</see>
891 </event>

893 <event id="AUE_role_logout" instance_of="AUE_generic_basic"
894   header="0" idNo="40" omit="JNI">
895   <title>su</title>
896   <program>/usr/bin/su</program>
897   <see>su(1M)</see>
898 </event>

900 <event id="AUE_newgrp_login" header="0" idNo="41" omit="JNI">
901   <program>newgrp</program>
902   <entry id="subject">
903     <internal token="subject"/>
904     <external opt="none"/>
905   </entry>
906   <entry id="groupname">
907     <internal token="text"/>
908     <external opt="required" type="char *"/>
909     <comment>group name</comment>
910   </entry>
911   <entry id="return">
912     <internal token="return"/>
913     <external opt="none"/>
914   </entry>
915 </event>

917 <event id="AUE_generic_mountable" type="generic" omit="always">
918   <!--

```

```

920      User device mounting related functions

922      -->
923      <entry id="subject">
924        <internal token="subject"/>
925        <external opt="none"/>
926      </entry>
927      <entry id="auth_used">
928        <internal token="uauth"/>
929        <external opt="required" type="char *"/>
930        <comment>authorization used</comment>
931      </entry>
932      <entry id="mount_point">
933        <internal token="path"/>
934        <external opt="required" type="char *"/>
935        <comment>mount point</comment>
936      </entry>
937      <entry id="device">
938        <internal token="path"/>
939        <external opt="required" type="char *"/>
940        <comment>device</comment>
941      </entry>
942      <entry id="options">
943        <internal token="text"/>
944        <external opt="optional" type="char *"/>
945        <comment>options</comment>
946      </entry>
947      <entry id="return">
948        <internal token="return"/>
949        <external opt="none"/>
950      </entry>
951    </event>

953    <event id="AUE_attach" instance_of="AUE_generic_moumountable"
954      header="0" idNo="42" omit="JNI">
955      <program>hald</program>
956    </event>
957    <event id="AUE_detach" instance_of="AUE_generic_moumountable"
958      header="0" idNo="43" omit="JNI">
959      <program>hald</program>
960    </event>
961    <event id="AUE_remove" header="0" idNo="44" omit="JNI">
962      <program>hald</program>
963      <entry id="subject">
964        <internal token="subject"/>
965        <external opt="none"/>
966      </entry>
967      <entry id="auth_used">
968        <internal token="uauth"/>
969        <external opt="required" type="char *"/>
970        <comment>authorization used</comment>
971      </entry>
972      <entry id="mount_point">
973        <internal token="path"/>
974        <external opt="optional" type="char *"/>
975        <comment>mount point</comment>
976      </entry>
977      <entry id="device">
978        <internal token="path"/>
979        <external opt="required" type="char *"/>
980        <comment>device</comment>
981      </entry>
982      <entry id="return">
983        <internal token="return"/>
984        <external opt="none"/>
985      </entry>

```

```

986    </event>

988    <event id="AUE_pool_import" header="0" idNo="45" omit="JNI">
989      <program>hald</program>
990      <entry id="subject">
991        <internal token="subject"/>
992        <external opt="none"/>
993      </entry>
994      <entry id="auth_used">
995        <internal token="uauth"/>
996        <external opt="required" type="char *"/>
997        <comment>authorization used</comment>
998      </entry>
999      <entry id="pool">
1000        <internal token="text"/>
1001        <external opt="required" type="char *"/>
1002        <comment>pool</comment>
1003      </entry>
1004      <entry id="device">
1005        <internal token="path"/>
1006        <external opt="required" type="char *"/>
1007        <comment>device</comment>
1008      </entry>
1009      <entry id="return">
1010        <internal token="return"/>
1011        <external opt="none"/>
1012      </entry>
1013    </event>
1014    <event id="AUE_pool_export" header="0" idNo="46" omit="JNI">
1015      <program>hald</program>
1016      <entry id="subject">
1017        <internal token="subject"/>
1018        <external opt="none"/>
1019      </entry>
1020      <entry id="auth_used">
1021        <internal token="uauth"/>
1022        <external opt="required" type="char *"/>
1023        <comment>authorization used</comment>
1024      </entry>
1025      <entry id="pool">
1026        <internal token="text"/>
1027        <external opt="required" type="char *"/>
1028        <comment>pool</comment>
1029      </entry>
1030      <entry id="device">
1031        <internal token="path"/>
1032        <external opt="required" type="char *"/>
1033        <comment>device</comment>
1034      </entry>
1035      <entry id="return">
1036        <internal token="return"/>
1037        <external opt="none"/>
1038      </entry>
1039    </event>

1041    <!-- dladm security objected events -->
1042    <event id="AUE_dladm_generic" type="generic" omit="always">
1043      <entry id="subject">
1044        <internal token="subject"/>
1045        <external opt="none"/>
1046      </entry>
1047      <entry id="auth_used">
1048        <internal token="uauth"/>
1049        <external opt="required" type="char *"/>
1050        <comment>authorization used</comment>
1051      </entry>

```

```

1052     <entry id="obj_class">
1053         <internal token="text"/>
1054         <external opt="required" type="char *"/>
1055         <comment>object class name</comment>
1056     </entry>
1057     <entry id="obj_name">
1058         <internal token="text"/>
1059         <external opt="required" type="char *"/>
1060         <comment>object name</comment>
1061     </entry>
1062     <entry id="return">
1063         <internal token="return"/>
1064         <external opt="none"/>
1065     </entry>
1066 </event>

1068 <event id="AUE_dladm_create_secobj" instance_of="AUE_dladm_generic"
1069     header="0" idNo="47" omit="JNI">
1070     <title>create wifi security object</title>
1071     <program>/usr/sbin/dladm</program>
1072     <see>dladm(1M)</see>
1073 </event>
1074 <event id="AUE_dladm_delete_secobj" instance_of="AUE_dladm_generic"
1075     header="0" idNo="48" omit="JNI">
1076     <title>delete wifi security object</title>
1077     <program>/usr/sbin/dladm</program>
1078     <see>dladm(1M)</see>
1079 </event>

1081 <!-- Trusted eXtensions (TX) events -->

1083 <!-- labeld events -->
1084 <event id="AUE_file_relabel" header="0" idNo="49" omit="JNI">
1085     <title>relabel file from one zone to another</title>
1086     <program>setlabel(1)</program>
1087     <see>setlabel(3TSOL)</see>
1088     <entry id="subject">
1089         <internal token="subject"/>
1090         <external opt="none"/>
1091     </entry>
1092     <entry id="auth_used">
1093         <internal token="uauth"/>
1094         <external opt="required" type="char *"/>
1095         <comment>authorization used</comment>
1096     </entry>
1097     <entry id="file">
1098         <internal token="path"/>
1099         <external opt="required" type="char *"/>
1100         <comment>file relabeled</comment>
1101     </entry>
1102     <entry id="src_label">
1103         <internal token="label"/>
1104         <external opt="required" type="m_label_t *"/>
1105         <comment>original label</comment>
1106     </entry>
1107     <entry id="dst_label">
1108         <internal token="label"/>
1109         <external opt="required" type="m_label_t *"/>
1110         <comment>new label</comment>
1111     </entry>
1112     <entry id="return">
1113         <internal token="return"/>
1114         <external opt="none"/>
1115     </entry>
1116 </event>

```

```

1118 <event id="AUE_file_copy" header="0" idNo="50" omit="JNI">
1119     <title>copy file to another zone</title>
1120     <program>dtfile(1X)</program>
1121     <entry id="subject">
1122         <internal token="subject"/>
1123         <external opt="none"/>
1124     </entry>
1125     <entry id="auth_used">
1126         <internal token="uauth"/>
1127         <external opt="required" type="char *"/>
1128         <comment>authorization used</comment>
1129     </entry>
1130     <entry id="src_file">
1131         <internal token="path"/>
1132         <external opt="required" type="char *"/>
1133         <comment>source file</comment>
1134     </entry>
1135     <entry id="src_label">
1136         <internal token="label"/>
1137         <external opt="required" type="m_label_t *"/>
1138         <comment>source label</comment>
1139     </entry>
1140     <entry id="dst_file">
1141         <internal token="path"/>
1142         <external opt="required" type="char *"/>
1143         <comment>destination directory</comment>
1144     </entry>
1145     <entry id="dst_label">
1146         <internal token="label"/>
1147         <external opt="required" type="m_label_t *"/>
1148         <comment>destination label</comment>
1149     </entry>
1150     <entry id="return">
1151         <internal token="return"/>
1152         <external opt="none"/>
1153     </entry>
1154 </event>

1156 <!-- uadmin(1m) events -->
1157 <event id="AUE_uadmin_generic" type="generic" omit="always">
1158     <entry id="subject">
1159         <internal token="subject"/>
1160         <external opt="none"/>
1161     </entry>
1162     <entry id="fcn">
1163         <internal token="text"/>
1164         <external opt="required" type="msg uadmin_fcn"/>
1165         <comment>next action</comment>
1166     </entry>
1167     <entry id="mdep">
1168         <internal token="text"/>
1169         <external opt="optional" type="char *"/>
1170         <comment>machine dependent argument</comment>
1171     </entry>
1172     <entry id="return">
1173         <internal token="return"/>
1174         <external opt="none"/>
1175     </entry>
1176 </event>
1177 <event id="AUE_uadmin_generic_fcn" type="generic" omit="always">
1178     <entry id="subject">
1179         <internal token="subject"/>
1180         <external opt="none"/>
1181     </entry>
1182     <entry id="fcn">
1183         <internal token="text"/>

```

```

1184         <external opt="required" type="msg uadmin_fcn"/>
1185         <comment>next action</comment>
1186     </entry>
1187     <entry id="return">
1188         <internal token="return"/>
1189         <external opt="none"/>
1190     </entry>
1191 </event>
1192 <event id="AUE_uadmin_shutdown" instance_of="AUE_uadmin_generic"
1193     header="0" idNo="51" omit="JNI">
1194     <title>uadmin shutdown</title>
1195     <program>/sbin/uadmin</program>
1196     <program>/usr/sbin/uadmin</program>
1197     <see>uadmin(1M)</see>
1198 </event>
1199 <event id="AUE_uadmin_reboot" instance_of="AUE_uadmin_generic"
1200     header="0" idNo="52" omit="JNI">
1201     <title>uadmin reboot</title>
1202     <program>/sbin/uadmin</program>
1203     <program>/usr/sbin/uadmin</program>
1204     <see>uadmin(1M)</see>
1205 </event>
1206 <event id="AUE_uadmin_dump" instance_of="AUE_uadmin_generic"
1207     header="0" idNo="53" omit="JNI">
1208     <title>uadmin dump</title>
1209     <program>/sbin/uadmin</program>
1210     <program>/usr/sbin/uadmin</program>
1211     <see>uadmin(1M)</see>
1212 </event>
1213 <event id="AUE_uadmin_freeze" instance_of="AUE_uadmin_generic"
1214     header="0" idNo="54" omit="JNI">
1215     <title>uadmin freeze</title>
1216     <program>/sbin/uadmin</program>
1217     <program>/usr/sbin/uadmin</program>
1218     <see>uadmin(1M)</see>
1219 </event>
1220 <event id="AUE_uadmin_remount" header="0" idNo="55" omit="JNI">
1221     <title>uadmin remount</title>
1222     <program>/sbin/uadmin</program>
1223     <program>/usr/sbin/uadmin</program>
1224     <see>uadmin(1M)</see>
1225     <entry id="subject">
1226         <internal token="subject"/>
1227         <external opt="none"/>
1228     </entry>
1229     <entry id="return">
1230         <internal token="return"/>
1231         <external opt="none"/>
1232     </entry>
1233 </event>
1234 <!-- uadmin ftrace and swapctl are not documented in uadmin(2) -->
1235 <event id="AUE_uadmin_ftrace" instance_of="AUE_uadmin_generic"
1236     header="0" idNo="56" omit="JNI">
1237     <title>uadmin ftrace</title>
1238     <program>/sbin/uadmin</program>
1239     <program>/usr/sbin/uadmin</program>
1240     <see>uadmin(1M)</see>
1241 </event>
1242 <event id="AUE_uadmin_swapctl" instance_of="AUE_uadmin_generic_fcn"
1243     header="0" idNo="57" omit="JNI">
1244     <title>uadmin swapctl</title>
1245     <program>/sbin/uadmin</program>
1246     <program>/usr/sbin/uadmin</program>
1247     <see>uadmin(1M)</see>
1248 </event>
1249 <event id="AUE_uadmin_thaw" header="0" idNo="96" omit="JNI">

```

```

1250     <title>thaw after freeze</title>
1251     <program>/sbin/uadmin</program>
1252     <program>/usr/sbin/uadmin</program>
1253     <see>uadmin(1M)</see>
1254     <entry id="subject">
1255         <internal token="subject"/>
1256         <external opt="none"/>
1257     </entry>
1258     <entry id="fcn">
1259         <internal token="text"/>
1260         <external opt="required" type="msg uadmin_fcn"/>
1261         <comment>freeze action type</comment>
1262     </entry>
1263     <entry id="return">
1264         <internal token="return"/>
1265         <external opt="none"/>
1266     </entry>
1267 </event>
1268 <!-- uadmin config is not documented in uadmin(2) -->
1269 <event id="AUE_uadmin_config" instance_of="AUE_uadmin_generic"
1270     header="0" idNo="119" omit="JNI">
1271     <title>uadmin config</title>
1272     <program>/sbin/uadmin</program>
1273     <program>/usr/sbin/uadmin</program>
1274     <see>uadmin(1M)</see>
1275 </event>
1276
1277 <!-- smbd service event; smbd session setup -->
1278 <event id="AUE_smbd_session" header="0" idNo="58" omit="JNI">
1279     <title>smbd</title>
1280     <program>/usr/lib/smbd/smbd</program>
1281     <entry id="subject">
1282         <internal token="subject"/>
1283         <external opt="none"/>
1284     </entry>
1285     <entry id="domain">
1286         <internal token="text"/>
1287         <external opt="required" type="char*" />
1288     <comment>domain</comment>
1289     </entry>
1290     <entry id="username">
1291         <internal token="text"/>
1292         <external opt="required" type="char*" />
1293     <comment>username</comment>
1294     </entry>
1295     <entry id="sid">
1296         <internal token="text"/>
1297         <external opt="optional" type="char*" />
1298     <comment>sid</comment>
1299     </entry>
1300     <entry id="return">
1301         <internal token="return"/>
1302         <external opt="none"/>
1303     </entry>
1304 </event>
1305
1306 <!-- smbd service event; smbd session logoff -->
1307 <event id="AUE_smbd_logoff" header="0" idNo="59" omit="JNI">
1308     <title>smbd</title>
1309     <program>/usr/lib/smbd/smbd</program>
1310     <entry id="subject">
1311         <internal token="subject"/>
1312         <external opt="none"/>
1313     </entry>
1314     <entry id="domain">
1315         <internal token="text"/>

```

```

1316         <external opt="required" type="char*" />
1317         <comment>domain</comment>
1318     </entry>
1319     <entry id="username">
1320         <internal token="text" />
1321         <external opt="required" type="char*" />
1322         <comment>username</comment>
1323     </entry>
1324     <entry id="return">
1325         <internal token="return" />
1326         <external opt="none" />
1327     </entry>
1328 </event>

1330 <!-- vscan service event; infected file detected -->
1331 <event id="AUE_vscan_quarantine" header="0" idNo="60" omit="JNI">
1332     <title>VSCAN: quarantine infected file</title>
1333     <program>/usr/lib/vscan/vscand</program>
1334     <see>vscand(1M), ICAP RFC 3507 (Extensions)</see>
1335     <entry id="subject">
1336         <internal token="subject" />
1337         <external opt="none" />
1338     </entry>
1339     <entry id="file">
1340         <internal token="path" />
1341         <external opt="required" type="char*" />
1342         <comment>infected file</comment>
1343     </entry>
1344     <entry id="violations,nviolations">
1345         <internal token="text" />
1346         <external opt="optional" type="char**,int" />
1347         <comment>ID - threat description</comment>
1348     </entry>
1349     <entry id="return">
1350         <internal token="return" />
1351         <external opt="none" />
1352     </entry>
1353 </event>

1355 <!-- ndmp service event; ndmp client connect -->
1356 <event id="AUE_ndmp_connect" instance_of="AUE_generic_basic" header="0"
1357     idNo="61" omit="JNI">
1358     <title>NDMP Connect</title>
1359     <program>/usr/lib/ndmp/ndmpd</program>
1360     <see>ndmpd(1M)</see>
1361 </event>

1363 <!-- ndmp service event; ndmp client disconnect -->
1364 <event id="AUE_ndmp_disconnect" instance_of="AUE_generic_basic" header="0"
1365     idNo="62" omit="JNI">
1366     <title>NDMP Disconnect</title>
1367     <program>/usr/lib/ndmp/ndmpd</program>
1368     <see>ndmpd(1M)</see>
1369 </event>

1371 <!-- ndmp service event; ndmp backup -->
1372 <event id="AUE_ndmp_backup" header="0" idNo="63" omit="JNI">
1373     <title>NDMP Backup</title>
1374     <program>/usr/lib/ndmp/ndmpd</program>
1375     <see>ndmpd(1M)</see>
1376     <entry id="subject">
1377         <internal token="subject" />
1378         <external opt="none" />
1379     </entry>
1380     <entry id="source">
1381         <internal token="path" />

```

```

1382         <external opt="required" type="char *" />
1383         <comment>path to be backed up</comment>
1384     </entry>
1385     <entry id="local_dest">
1386         <internal token="path" />
1387         <external opt="optional" type="char *" />
1388         <comment>local path of backup destination</comment>
1389     </entry>
1390     <entry id="remote_dest">
1391         <internal token="in_peer" />
1392         <external opt="optional" type="fd_t" />
1393         <comment>remote ip address and port of backup destination</comment>
1394     </entry>
1395     <entry id="return">
1396         <internal token="return" />
1397         <external opt="none" />
1398     </entry>
1399 </event>

1401 <!-- ndmp service event; ndmp restore -->
1402 <event id="AUE_ndmp_restore" header="0" idNo="64" omit="JNI">
1403     <title>NDMP Restore</title>
1404     <program>/usr/lib/ndmp/ndmpd</program>
1405     <see>ndmpd(1M)</see>
1406     <entry id="subject">
1407         <internal token="subject" />
1408         <external opt="none" />
1409     </entry>
1410     <entry id="destination">
1411         <internal token="path" />
1412         <external opt="required" type="char *" />
1413         <comment>path to restore to</comment>
1414     </entry>
1415     <entry id="local_source">
1416         <internal token="path" />
1417         <external opt="optional" type="char *" />
1418         <comment>local path to restore from</comment>
1419     </entry>
1420     <entry id="remote_source">
1421         <internal token="in_peer" />
1422         <external opt="optional" type="fd_t" />
1423         <comment>remote ip address and port to restore from</comment>
1424     </entry>
1425     <entry id="return">
1426         <internal token="return" />
1427         <external opt="none" />
1428     </entry>
1429 </event>

1431 <!-- SMF related events -->
1432 <event id="AUE_smf_generic" type="generic" omit="always">
1433     <!--
1434     This is a template for the event types that have no tokens
1435     other than the header and return. There is no allowed_type
1436     list because the template is not externally visible due to the
1437     omit="always".
1438     -->
1439     <entry id="subject">
1440         <internal token="subject" />
1441         <external opt="none" />
1442     </entry>
1443     <entry id="auth_used">
1444         <internal token="uauth" />
1445         <external opt="required" type="char *" />
1446         <comment>authorization used</comment>
1447 </entry>

```



```

1448     <entry id="fmri">
1449         <internal token="fmri"/>
1450         <external opt="required" type="char *"/>
1451         <comment>name</comment>
1452     </entry>
1453     <entry id="return">
1454         <internal token="return"/>
1455         <external opt="none"/>
1456     </entry>
1457 </event>

1459 <event id="AUE_smf_generic_pg" type="generic" omit="always">
1460     <!--
1461     This is a template for the event types related to property groups.
1462     There is no allowed_type list because the template is not externally
1463     visible due to the omit="always".
1464     -->
1465     <entry id="subject">
1466         <internal token="subject"/>
1467         <external opt="none"/>
1468     </entry>
1469     <entry id="auth_used">
1470         <internal token="uauth"/>
1471         <external opt="required" type="char *"/>
1472         <comment>authorization used</comment>
1473     </entry>
1474     <entry id="fmri">
1475         <internal token="fmri"/>
1476         <external opt="required" type="char *"/>
1477     </entry>
1478     <entry id="type">
1479         <internal token="text"/>
1480         <external opt="required" type="char *"/>
1481         <comment>property group type</comment>
1482     </entry>
1483     <entry id="return">
1484         <internal token="return"/>
1485         <external opt="none"/>
1486     </entry>
1487 </event>

1489 <event id="AUE_smf_enable" instance_of="AUE_smf_generic" header="0"
1490     idNo="65" omit="JNI">
1491     <program>svc.configd(1M)</program>
1492     <see>svcadm(1M)</see>
1493 </event>
1494 <event id="AUE_smf_tmp_enable" instance_of="AUE_smf_generic" header="0"
1495     idNo="66" omit="JNI">
1496     <program>svc.configd(1M)</program>
1497     <see>svcadm(1M)</see>
1498 </event>
1499 <event id="AUE_smf_disable" instance_of="AUE_smf_generic" header="0"
1500     idNo="67" omit="JNI">
1501     <program>svc.configd(1M)</program>
1502     <see>svcadm(1M)</see>
1503 </event>
1504 <event id="AUE_smf_tmp_disable" instance_of="AUE_smf_generic" header="0"
1505     idNo="68" omit="JNI">
1506     <program>svc.configd(1M)</program>
1507     <see>svcadm(1M)</see>
1508 </event>
1509 <event id="AUE_smf_restart" instance_of="AUE_smf_generic" header="0"
1510     idNo="69" omit="JNI">
1511     <program>svc.configd(1M)</program>
1512     <see>svcadm(1M)</see>
1513 </event>

```

```

1514 <event id="AUE_smf_refresh" instance_of="AUE_smf_generic" header="0"
1515     idNo="70" omit="JNI">
1516     <program>svc.configd(1M)</program>
1517     <see>svcadm(1M)</see>
1518 </event>
1519 <event id="AUE_smf_clear" instance_of="AUE_smf_generic" header="0"
1520     idNo="71" omit="JNI">
1521     <program>svc.configd(1M)</program>
1522     <see>svcadm(1M)</see>
1523 </event>
1524 <event id="AUE_smf_degrade" instance_of="AUE_smf_generic" header="0"
1525     idNo="72" omit="JNI">
1526     <program>svc.configd(1M)</program>
1527     <see>svcadm(1M)</see>
1528 </event>
1529 <event id="AUE_smf_immediate_degrade" instance_of="AUE_smf_generic"
1530     header="0" idNo="73" omit="JNI">
1531     <program>svc.configd(1M)</program>
1532     <see>svcadm(1M)</see>
1533 </event>
1534 <event id="AUE_smf_maintenance" instance_of="AUE_smf_generic" header="0"
1535     idNo="74" omit="JNI">
1536     <program>svc.configd(1M)</program>
1537     <see>svcadm(1M)</see>
1538 </event>
1539 <event id="AUE_smf_immediate_maintenance" instance_of="AUE_smf_generic"
1540     header="0" idNo="75" omit="JNI">
1541     <program>svc.configd(1M)</program>
1542     <see>svcadm(1M)</see>
1543 </event>
1544 <event id="AUE_smf_immtmp_maintenance" instance_of="AUE_smf_generic"
1545     header="0" idNo="76" omit="JNI">
1546     <program>svc.configd(1M)</program>
1547     <see>svcadm(1M)</see>
1548 </event>
1549 <event id="AUE_smf_tmp_maintenance" instance_of="AUE_smf_generic" header="0"
1550     idNo="77" omit="JNI">
1551     <program>svc.configd(1M)</program>
1552     <see>svcadm(1M)</see>
1553 </event>
1554 <event id="AUE_smf_milestone" instance_of="AUE_smf_generic" header="0"
1555     idNo="78" omit="JNI">
1556     <program>svc.configd(1M)</program>
1557     <see>svcadm(1M)</see>
1558 </event>

1560 <event id="AUE_smf_create" instance_of="AUE_smf_generic" header="0"
1561     idNo="79" omit="JNI">
1562     <program>svc.configd(1M)</program>
1563     <see>svccfg(1M)</see>
1564 </event>
1565 <event id="AUE_smf_delete" instance_of="AUE_smf_generic" header="0"
1566     idNo="80" omit="JNI">
1567     <program>svc.configd(1M)</program>
1568     <see>svccfg(1M)</see>
1569 </event>

1571 <event id="AUE_smf_create_pg" instance_of="AUE_smf_generic_pg" header="0"
1572     idNo="81" omit="JNI">
1573     <program>svc.configd(1M)</program>
1574     <see>svccfg(1M)</see>
1575 </event>
1576 <event id="AUE_smf_create_npg" instance_of="AUE_smf_generic_pg" header="0"
1577     idNo="82" omit="JNI">
1578     <program>svc.configd(1M)</program>
1579     <see>svccfg(1M)</see>

```

```

1580 </event>
1581 <event id="AUE_smf_delete_pg" instance_of="AUE_smf_generic_pg" header="0"
1582       idNo="83" omit="JNI">
1583   <program>svc.configd(1M)</program>
1584   <see>svccfg(1M)</see>
1585 </event>
1586 <event id="AUE_smf_delete_npg" instance_of="AUE_smf_generic_pg" header="0"
1587       idNo="84" omit="JNI">
1588   <program>svc.configd(1M)</program>
1589   <see>svccfg(1M)</see>
1590 </event>
1592 <event id="AUE_smf_create_snap" header="0" idNo="85" omit="JNI">
1593   <program>svc.configd(1M)</program>
1594   <see>svccfg(1M)</see>
1595   <entry id="subject">
1596     <internal token="subject"/>
1597     <external opt="none"/>
1598   </entry>
1599   <entry id="auth_used">
1600     <internal token="uauth"/>
1601     <external opt="required" type="char *"/>
1602     <comment>authorization used</comment>
1603   </entry>
1604   <entry id="fmri">
1605     <internal token="fmri"/>
1606     <external opt="required" type="char *"/>
1607     <comment>name</comment>
1608   </entry>
1609   <entry id="name">
1610     <internal token="text"/>
1611     <external opt="required" type="char *"/>
1612     <comment>snapshot name</comment>
1613   </entry>
1614   <entry id="return">
1615     <internal token="return"/>
1616     <external opt="none"/>
1617   </entry>
1618 </event>
1619 <event id="AUE_smf_delete_snap" header="0" idNo="86" omit="JNI">
1620   <program>svc.configd(1M)</program>
1621   <see>svccfg(1M)</see>
1622   <entry id="subject">
1623     <internal token="subject"/>
1624     <external opt="none"/>
1625   </entry>
1626   <entry id="auth_used">
1627     <internal token="uauth"/>
1628     <external opt="required" type="char *"/>
1629     <comment>authorization used</comment>
1630   </entry>
1631   <entry id="fmri">
1632     <internal token="fmri"/>
1633     <external opt="required" type="char *"/>
1634     <comment>name</comment>
1635   </entry>
1636   <entry id="name">
1637     <internal token="text"/>
1638     <external opt="required" type="char *"/>
1639     <comment>snapshot name</comment>
1640   </entry>
1641   <entry id="return">
1642     <internal token="return"/>
1643     <external opt="none"/>
1644   </entry>
1645 </event>

```

```

1646 <event id="AUE_smf_attach_snap" header="0" idNo="87" omit="JNI">
1647   <program>svc.configd(1M)</program>
1648   <see>svccfg(1M)</see>
1649   <entry id="subject">
1650     <internal token="subject"/>
1651     <external opt="none"/>
1652   </entry>
1653   <entry id="auth_used">
1654     <internal token="uauth"/>
1655     <external opt="required" type="char *"/>
1656     <comment>authorization used</comment>
1657   </entry>
1658   <entry id="old_fmri">
1659     <internal token="fmri"/>
1660     <external opt="required" type="char *"/>
1661     <comment>old name</comment>
1662   </entry>
1663   <entry id="old_name">
1664     <internal token="text"/>
1665     <external opt="required" type="char *"/>
1666     <comment>old snapshot</comment>
1667   </entry>
1668   <entry id="new_fmri">
1669     <internal token="fmri"/>
1670     <external opt="required" type="char *"/>
1671     <comment>new name</comment>
1672   </entry>
1673   <entry id="new_name">
1674     <internal token="text"/>
1675     <external opt="required" type="char *"/>
1676     <comment>new snapshot</comment>
1677   </entry>
1678   <entry id="return">
1679     <internal token="return"/>
1680     <external opt="none"/>
1681   </entry>
1682 </event>
1684 <event id="AUE_smf_annotation" header="0" idNo="88" omit="JNI">
1685   <program>svc.configd(1M)</program>
1686   <see>svccfg(1M)</see>
1687   <entry id="subject">
1688     <internal token="subject"/>
1689     <external opt="none"/>
1690   </entry>
1691   <entry id="operation">
1692     <internal token="text"/>
1693     <external opt="required" type="char *"/>
1694     <comment>operation</comment>
1695   </entry>
1696   <entry id="file">
1697     <internal token="path"/>
1698     <external opt="required" type="char *"/>
1699     <comment>imported file</comment>
1700   </entry>
1701   <entry id="return">
1702     <internal token="return"/>
1703     <external opt="none"/>
1704   </entry>
1705 </event>
1707 <event id="AUE_smf_create_prop" header="0" idNo="89" omit="JNI">
1708   <program>svc.configd(1M)</program>
1709   <see>svccfg(1M)</see>
1710   <entry id="subject">
1711     <internal token="subject"/>

```

```

1712     <external opt="none"/>
1713 </entry>
1714 <entry id="auth_used">
1715   <internal token="uauth"/>
1716   <external opt="required" type="char *"/>
1717   <comment>authorization used</comment>
1718 </entry>
1719 <entry id="fmri">
1720   <internal token="fmri"/>
1721   <external opt="required" type="char *"/>
1722   <comment>name</comment>
1723 </entry>
1724 <entry id="type">
1725   <internal token="text"/>
1726   <external opt="required" type="char *"/>
1727   <comment>type</comment>
1728 </entry>
1729 <entry id="value">
1730   <internal token="text"/>
1731   <external opt="optional" type="char *"/>
1732   <comment>value</comment>
1733 </entry>
1734 <entry id="return">
1735   <internal token="return"/>
1736   <external opt="none"/>
1737 </entry>
1738 </event>

1740 <event id="AUE_smf_change_prop" header="0" idNo="90" omit="JNI">
1741   <program>svc.configd(1M)</program>
1742   <see>svccfg(1M)</see>
1743   <entry id="subject">
1744     <internal token="subject"/>
1745     <external opt="none"/>
1746   </entry>
1747   <entry id="auth_used">
1748     <internal token="uauth"/>
1749     <external opt="required" type="char *"/>
1750     <comment>authorization used</comment>
1751   </entry>
1752   <entry id="fmri">
1753     <internal token="fmri"/>
1754     <external opt="required" type="char *"/>
1755     <comment>name</comment>
1756   </entry>
1757   <entry id="type">
1758     <internal token="text"/>
1759     <external opt="required" type="char *"/>
1760     <comment>type</comment>
1761   </entry>
1762   <entry id="value">
1763     <internal token="text"/>
1764     <external opt="optional" type="char *"/>
1765     <comment>value</comment>
1766   </entry>
1767   <entry id="return">
1768     <internal token="return"/>
1769     <external opt="none"/>
1770   </entry>
1771 </event>
1772 <event id="AUE_smf_delete_prop" header="0" idNo="91" omit="JNI">
1773   <program>svc.configd(1M)</program>
1774   <see>svccfg(1M)</see>
1775   <entry id="subject">
1776     <internal token="subject"/>
1777     <external opt="none"/>

```

```

1778 </entry>
1779 <entry id="auth_used">
1780   <internal token="uauth"/>
1781   <external opt="required" type="char *"/>
1782   <comment>authorization used</comment>
1783 </entry>
1784 <entry id="fmri">
1785   <internal token="fmri"/>
1786   <external opt="required" type="char *"/>
1787   <comment>name</comment>
1788 </entry>
1789 <entry id="return">
1790   <internal token="return"/>
1791   <external opt="none"/>
1792 </entry>
1793 </event>

1795 <event id="AUE_smf_read_prop" instance_of="AUE_smf_generic" header="0"
1796   idNo="92" omit="JNI">
1797   <program>svc.configd(1M)</program>
1798   <see>svccfg(1M)</see>
1799 </event>

1801 <!-- CPUFreq related events -->

1803 <event id="AUE_cpu_ondemand" header="0" idNo="93" omit="JNI">
1804   <title>set CPU freq to minimal unless load increases</title>
1805   <program>/usr/lib/hal/hald-addon-cpufreq</program>
1806   <see>hald(1M)</see>
1807   <entry id="subject">
1808     <internal token="subject"/>
1809     <external opt="none"/>
1810   </entry>
1811   <entry id="auth_used">
1812     <internal token="uauth"/>
1813     <external opt="required" type="char *"/>
1814     <comment>authorization used</comment>
1815   </entry>
1816   <entry id="return">
1817     <internal token="return"/>
1818     <external opt="none"/>
1819   </entry>
1820 </event>
1821 <event id="AUE_cpu_performance" header="0" idNo="94" omit="JNI">
1822   <title>set CPU freq to Max</title>
1823   <program>/usr/lib/hal/hald-addon-cpufreq</program>
1824   <see>hald(1M)</see>
1825   <entry id="subject">
1826     <internal token="subject"/>
1827     <external opt="none"/>
1828   </entry>
1829   <entry id="auth_used">
1830     <internal token="uauth"/>
1831     <external opt="required" type="char *"/>
1832     <comment>authorization used</comment>
1833   </entry>
1834   <entry id="return">
1835     <internal token="return"/>
1836     <external opt="none"/>
1837   </entry>
1838 </event>
1839 <event id="AUE_cpu_threshold" header="0" idNo="95" omit="JNI">
1840   <title>set CPU frequency threshold percentage</title>
1841   <program>/usr/lib/hal/hald-addon-cpufreq</program>
1842   <see>hald(1M)</see>
1843   <entry id="subject">

```

```

1844     <internal token="subject"/>
1845     <external opt="none"/>
1846   </entry>
1847   <entry id="auth_used">
1848     <internal token="uauth"/>
1849     <external opt="required" type="char *"/>
1850     <comment>authorization used</comment>
1851   </entry>
1852   <entry id="threshold">
1853     <internal token="text"/>
1854     <external opt="required" type="int"/>
1855     <comment>threshold percent 1-100</comment>
1856   </entry>
1857   <entry id="return">
1858     <internal token="return"/>
1859     <external opt="none"/>
1860   </entry>
1861 </event>

```

1863 <!-- TPM events recorded by tcscd(8) -->

```

1865 <event id="AUE_generic_tpm" type="generic" omit="always">
1866   <entry id="subject">
1867     <internal token="subject"/>
1868     <external opt="none"/>
1869   </entry>
1870   <entry id="message">
1871     <internal token="text"/>
1872     <external opt="optional" type="msg tpm_e"/>
1873     <comment>TPM error message</comment>
1874   </entry>
1875   <entry id="return">
1876     <internal token="return"/>
1877     <external opt="none"/>
1878   </entry>
1879 </event>

1881 <event id="AUE_tpm_takeownership" instance_of="AUE_generic_tpm"
1882   header="0" idNo="99" omit="JNI">
1883   <title>TPM_TakeOwnership</title>
1884   <program>/usr/lib/tcsd</program>
1885   <see>tcscd(8)</see>
1886 </event>
1887 <event id="AUE_tpm_setoperatorauth" instance_of="AUE_generic_tpm"
1888   header="0" idNo="100" omit="JNI">
1889   <title>TPM_SetOperatorAuth</title>
1890   <program>/usr/lib/tcsd</program>
1891   <see>tcscd(8)</see>
1892 </event>
1893 <event id="AUE_tpm_setownerinstall" instance_of="AUE_generic_tpm"
1894   header="0" idNo="101" omit="JNI">
1895   <title>TPM_SetOwnerInstall</title>
1896   <program>/usr/lib/tcsd</program>
1897   <see>tcscd(8)</see>
1898 </event>
1899 <event id="AUE_tpm_selftestfull" instance_of="AUE_generic_tpm"
1900   header="0" idNo="102" omit="JNI">
1901   <title>TPM_SelfTestFull</title>
1902   <program>/usr/lib/tcsd</program>
1903   <see>tcscd(8)</see>
1904 </event>
1905 <event id="AUE_tpm_certifyselftest" instance_of="AUE_generic_tpm"
1906   header="0" idNo="103" omit="JNI">
1907   <title>TPM_CertifySelfTest</title>
1908   <program>/usr/lib/tcsd</program>
1909   <see>tcscd(8)</see>

```

```

1910 </event>
1911 <event id="AUE_tpm_continueselftest" instance_of="AUE_generic_tpm"
1912   header="0" idNo="104" omit="JNI">
1913   <title>TPM_ContinueSelfTest</title>
1914   <program>/usr/lib/tcsd</program>
1915   <see>tcscd(8)</see>
1916 </event>
1917 <event id="AUE_tpm_ownerasetdisable" instance_of="AUE_generic_tpm"
1918   header="0" idNo="105" omit="JNI">
1919   <title>TPM_OwnerSetDisable</title>
1920   <program>/usr/lib/tcsd</program>
1921   <see>tcscd(8)</see>
1922 </event>
1923 <event id="AUE_tpm_ownerclear" instance_of="AUE_generic_tpm"
1924   header="0" idNo="106" omit="JNI">
1925   <title>TPM_OwnerClear</title>
1926   <program>/usr/lib/tcsd</program>
1927   <see>tcscd(8)</see>
1928 </event>
1929 <event id="AUE_tpm_disableownerclear" instance_of="AUE_generic_tpm"
1930   header="0" idNo="107" omit="JNI">
1931   <title>TPM_DisableOwnerClear</title>
1932   <program>/usr/lib/tcsd</program>
1933   <see>tcscd(8)</see>
1934 </event>
1935 <event id="AUE_tpm_forceclear" instance_of="AUE_generic_tpm"
1936   header="0" idNo="108" omit="JNI">
1937   <title>TPM_ForceClear</title>
1938   <program>/usr/lib/tcsd</program>
1939   <see>tcscd(8)</see>
1940 </event>
1941 <event id="AUE_tpm_disableforceclear" instance_of="AUE_generic_tpm"
1942   header="0" idNo="109" omit="JNI">
1943   <title>TPM_DisableForceClear</title>
1944   <program>/usr/lib/tcsd</program>
1945   <see>tcscd(8)</see>
1946 </event>
1947 <event id="AUE_tpm_physicaldisable" instance_of="AUE_generic_tpm"
1948   header="0" idNo="110" omit="JNI">
1949   <title>TPM_PhysicalDisable</title>
1950   <program>/usr/lib/tcsd</program>
1951   <see>tcscd(8)</see>
1952 </event>
1953 <event id="AUE_tpm_physicalenable" instance_of="AUE_generic_tpm"
1954   header="0" idNo="111" omit="JNI">
1955   <title>TPM_PhysicalEnable</title>
1956   <program>/usr/lib/tcsd</program>
1957   <see>tcscd(8)</see>
1958 </event>
1959 <event id="AUE_tpm_physicaldeactivate" instance_of="AUE_generic_tpm"
1960   header="0" idNo="112" omit="JNI">
1961   <title>TPM_PhysicalSetDeactivated</title>
1962   <program>/usr/lib/tcsd</program>
1963   <see>tcscd(8)</see>
1964 </event>
1965 <event id="AUE_tpm_settempdeactivated" instance_of="AUE_generic_tpm"
1966   header="0" idNo="113" omit="JNI">
1967   <title>TPM_SetTempDeactivated</title>
1968   <program>/usr/lib/tcsd</program>
1969   <see>tcscd(8)</see>
1970 </event>
1971 <event id="AUE_tpm_physicalpresence" instance_of="AUE_generic_tpm"
1972   header="0" idNo="114" omit="JNI">
1973   <title>TPM_PhysicalPresence</title>
1974   <program>/usr/lib/tcsd</program>
1975   <see>tcscd(8)</see>

```

```

1976 </event>
1977 <event id="AUE_tpm_fieldupgrade" instance_of="AUE_generic_tpm"
1978   header="0" idNo="115" omit="JNI">
1979   <title>TPM_FieldUpgrade</title>
1980   <program>/usr/lib/tcsd</program>
1981   <see>tcsd(8)</see>
1982 </event>
1983 <event id="AUE_tpm_resetlockvalue" instance_of="AUE_generic_tpm"
1984   header="0" idNo="116" omit="JNI">
1985   <title>TPM_ResetLockValue</title>
1986   <program>/usr/lib/tcsd</program>
1987   <see>tcsd(8)</see>
1988 </event>

1990 <!-- hotplug events recorded by hotplugd(1m) -->

1992 <event id="AUE_hotplug_state" header="0" idNo="117" omit="JNI">
1993   <title>change hotplug connection state</title>
1994   <program>/usr/lib/hotplugd</program>
1995   <see>hotplugd(1M)</see>
1996   <entry id="subject">
1997     <internal token="subject"/>
1998     <external opt="none"/>
1999   </entry>
2000   <entry id="auth_used">
2001     <internal token="uauth"/>
2002     <external opt="required" type="char **"/>
2003     <comment>authorization used</comment>
2004   </entry>
2005   <entry id="device_path">
2006     <internal token="path"/>
2007     <external opt="required" type="char **"/>
2008     <comment>device path</comment>
2009   </entry>
2010   <entry id="connection">
2011     <internal token="text"/>
2012     <external opt="required" type="char **"/>
2013     <comment>connector or port</comment>
2014   </entry>
2015   <entry id="new_state">
2016     <internal token="text"/>
2017     <external opt="required" type="char **"/>
2018     <comment>new connection state</comment>
2019   </entry>
2020   <entry id="old_state">
2021     <internal token="text"/>
2022     <external opt="required" type="char **"/>
2023     <comment>old connection state</comment>
2024   </entry>
2025   <entry id="return">
2026     <internal token="return"/>
2027     <external opt="none"/>
2028   </entry>
2029 </event>

2031 <event id="AUE_hotplug_set" header="0" idNo="118" omit="JNI">
2032   <title>set hotplug bus private options</title>
2033   <program>/usr/lib/hotplugd</program>
2034   <see>hotplugd(1M)</see>
2035   <entry id="subject">
2036     <internal token="subject"/>
2037     <external opt="none"/>
2038   </entry>
2039   <entry id="auth_used">
2040     <internal token="uauth"/>
2041     <external opt="required" type="char **"/>

```

```

2042     <comment>authorization used</comment>
2043   </entry>
2044   <entry id="device_path">
2045     <internal token="path"/>
2046     <external opt="required" type="char **"/>
2047     <comment>device path</comment>
2048   </entry>
2049   <entry id="connection">
2050     <internal token="text"/>
2051     <external opt="required" type="char **"/>
2052     <comment>connector or port</comment>
2053   </entry>
2054   <entry id="options">
2055     <internal token="text"/>
2056     <external opt="required" type="char **"/>
2057     <comment>bus private options</comment>
2058   </entry>
2059   <entry id="return">
2060     <internal token="return"/>
2061     <external opt="none"/>
2062   </entry>
2063 </event>

2065 <event id="AUE_ilb_create_healthcheck" header="0" idNo="120" omit="JNI">
2066   <title>Create Integrated Loadbalancer healthcheck object</title>
2067   <program>/usr/sbin/ilbadm</program>
2068   <see>ilbadm(1m)</see>
2069   <entry id="subject">
2070     <internal token="subject"/>
2071     <external opt="none"/>
2072   </entry>
2073   <entry id="auth_used">
2074     <internal token="uauth"/>
2075     <external opt="required" type="char **"/>
2076     <comment>authorization used</comment>
2077   </entry>
2078   <entry id="hc_test">
2079     <internal token="path"/>
2080     <external opt="required" type="char **"/>
2081     <comment>healthcheck type-PING,TCP,UDP or 3rd party script</comment>
2082   </entry>
2083   <entry id="hc_name">
2084     <internal token="text"/>
2085     <external opt="required" type="char **"/>
2086     <comment>healthcheck name</comment>
2087   </entry>
2088   <entry id="hc_timeout">
2089     <internal token="text"/>
2090     <external opt="required" type="int32_t" />
2091     <comment>timeout(secs) to kill a hung healthcheck probe
2092       - 0 means default value (see man page)
2093   </comment>
2094   </entry>
2095   <entry id="hc_count">
2096     <internal token="text"/>
2097     <external opt="required" type="int"/>
2098     <comment>number of times to run a health check probe
2099       before declaring a server to be dead - 0 means
2100       default value (see man page)
2101   </comment>
2102   </entry>
2103   <entry id="hc_interval">
2104     <internal token="text"/>
2105     <external opt="required" type="int32_t"/>
2106     <comment>time(secs) between 2 healthcheck events -
2107       0 means default value(see man page)

```

```

2108         </comment>
2109     </entry>
2110     <entry id="return">
2111         <internal token="return"/>
2112         <external opt="none"/>
2113     </entry>
2114 </event>

2116 <event id="AUE_ilb_delete_healthcheck" header="0" idNo="121" omit="JNI">
2117     <title>Delete Integrated Loadbalancer healthcheck object</title>
2118     <program>/usr/sbin/ilbadm</program>
2119     <see>ilbadm(1m)</see>
2120     <entry id="subject">
2121         <internal token="subject"/>
2122         <external opt="none"/>
2123     </entry>
2124     <entry id="auth_used">
2125         <internal token="uauth"/>
2126         <external opt="required" type="char *"/>
2127         <comment>authorization used</comment>
2128     </entry>
2129     <entry id="hc_name">
2130         <internal token="text"/>
2131         <external opt="required" type="char *"/>
2132         <comment>healthcheck name</comment>
2133     </entry>
2134     <entry id="return">
2135         <internal token="return"/>
2136         <external opt="none"/>
2137     </entry>
2138 </event>

2140 <event id="AUE_ilb_create_rule" header="0" idNo="122" omit="JNI">
2141     <title>Create Integrated Loadbalancer rule</title>
2142     <program>/usr/sbin/ilbadm</program>
2143     <see>ilbadm(1m)</see>
2144     <entry id="subject">
2145         <internal token="subject"/>
2146         <external opt="none"/>
2147     </entry>
2148     <entry id="auth_used">
2149         <internal token="uauth"/>
2150         <external opt="required" type="char *"/>
2151         <comment>authorization used</comment>
2152     </entry>
2153     <entry id="virtual_ipaddress_type,virtual_ipaddress">
2154         <internal token="in_remote"/>
2155         <external opt="required" type="int32_t,uint32_t[4]"/>
2156         <comment>LB virtual IP address</comment>
2157     </entry>
2158     <entry id="min_port">
2159         <internal token="iport"/>
2160         <external opt="required" type="uint16_t"/>
2161         <comment>minimum value in port range</comment>
2162     </entry>
2163     <entry id="max_port">
2164         <internal token="iport"/>
2165         <external opt="required" type="uint16_t"/>
2166         <comment>maximum value in port range - max=min means single
2167             port is specified
2168         </comment>
2169     </entry>
2170     <entry id="protocol">
2171         <internal token="text"/>
2172         <external opt="required" type="char *"/>
2173         <comment>protocol</comment>

```

```

2174     </entry>
2175     <entry id="algo_optype">
2176         <internal token="text"/>
2177         <external opt="required" type="char *"/>
2178         <comment>[rr,hip,hipp,hipv],[dsr,nat,half-nat]</comment>
2179     </entry>
2180     <entry id="proxy_src_min_type,proxy_src_min">
2181         <internal token="in_remote"/>
2182         <external opt="optional" type="int32_t,uint32_t[4]"/>
2183         <comment>min value for proxy source address for NAT</comment>
2184     </entry>
2185     <entry id="proxy_src_max_type,proxy_src_max">
2186         <internal token="in_remote"/>
2187         <external opt="optional" type="int32_t,uint32_t[4]"/>
2188         <comment>max value in proxy source address range for NAT
2189             - max=min means single address is specified
2190         </comment>
2191     </entry>
2192     <entry id="persist_mask">
2193         <internal token="text"/>
2194         <external opt="required" type="char *"/>
2195         <comment>prefix length</comment>
2196     </entry>
2197     <entry id="hcname">
2198         <internal token="text"/>
2199         <external opt="optional" type="char *"/>
2200         <comment>healthcheck name</comment>
2201     </entry>
2202     <entry id="hcport">
2203         <internal token="text"/>
2204         <external opt="optional" type="char *"/>
2205         <comment>healthcheck port - ANY(dynamically determined by ilbd)
2206             or a positive integer
2207         </comment>
2208     </entry>
2209     <entry id="conndrain_timeout">
2210         <internal token="text"/>
2211         <external opt="required" type="uint32_t"/>
2212         <comment>connection timeout for NAT/half-NAT in sec. - 0 means
2213             no forced removal)
2214         </comment>
2215     </entry>
2216     <entry id="nat_timeout">
2217         <internal token="text"/>
2218         <external opt="required" type="uint32_t"/>
2219         <comment>nat entry timeout for NAT/half-NAT in sec - 0 means
2220             default value(see man page)
2221         </comment>
2222     </entry>
2223     <entry id="persist_timeout">
2224         <internal token="text"/>
2225         <external opt="required" type="uint32_t"/>
2226         <comment>session persistence mapping in sec - 0 means no
2227             persistence
2228         </comment>
2229     </entry>
2230     <entry id="server_group">
2231         <internal token="text"/>
2232         <external opt="required" type="char *"/>
2233         <comment>server group name</comment>
2234     </entry>
2235     <entry id="rule_name">
2236         <internal token="text"/>
2237         <external opt="required" type="char *"/>
2238         <comment>rule name</comment>
2239     </entry>

```

```

2240     <entry id="return">
2241         <internal token="return"/>
2242     </entry>
2243     </event>
2244
2246 <!-- generic ILB rule event -->
2247
2248 <event id="AUE_generic_ILB_rule" type="generic" omit="always">
2249     <entry id="subject">
2250         <internal token="subject"/>
2251     </entry>
2252     <entry id="auth_used">
2253         <internal token="uauth"/>
2254         <external opt="required" type="char *"/>
2255         <comment>authorization used</comment>
2256     </entry>
2257     <entry id="rule_name">
2258         <internal token="text"/>
2259         <external opt="required" type="char *"/>
2260         <comment>rule name - "all" means all rules</comment>
2261     </entry>
2262     <entry id="return">
2263         <internal token="return"/>
2264     </entry>
2265 </event>
2266
2269 <!-- instances of the ILB generic rule event. -->
2270 <event id="AUE_ilb_delete_rule" instance_of="AUE_generic_ILB_rule">
2271     <header>"0" idNo="123">
2272         <title>Delete Integrated Loadbalancer rule</title>
2273         <program>/usr/sbin/ilbadm</program>
2274         <see>ilbadm(1m)</see>
2275     </event>
2276
2277 <event id="AUE_ilb_disable_rule" instance_of="AUE_generic_ILB_rule">
2278     <header>"0" idNo="124">
2279         <title>Disable Integrated Loadbalancer rule</title>
2280         <program>/usr/sbin/ilbadm</program>
2281         <see>ilbadm(1m)</see>
2282     </event>
2283
2284 <event id="AUE_ilb_enable_rule" instance_of="AUE_generic_ILB_rule">
2285     <header>"0" idNo="125">
2286         <title>Enable Integrated Loadbalancer rule</title>
2287         <program>/usr/sbin/ilbadm</program>
2288         <see>ilbadm(1m)</see>
2289     </event>
2290
2291 <event id="AUE_ilb_add_server" header="0" idNo="126" omit="JNI">
2292     <title>Add server to Integrated Loadbalancer</title>
2293     <program>/usr/sbin/ilbadm</program>
2294     <see>ilbadm(1m)</see>
2295     <entry id="subject">
2296         <internal token="subject"/>
2297         <external opt="none"/>
2298     </entry>
2299     <entry id="auth_used">
2300         <internal token="uauth"/>
2301         <external opt="required" type="char *"/>
2302         <comment>authorization used</comment>
2303     </entry>
2304     <entry id="server_ipaddress_type, server_ipaddress">
2305         <internal token="in_remote"/>

```

```

2306         <external opt="required" type="int32_t, uint32_t[4]"/>
2307     </comment>IP address</comment>
2308 </entry>
2309 <entry id="server_id">
2310     <internal token="text"/>
2311     <external opt="optional" type="char *"/>
2312     <comment>serverid that corresponds IP address - empty
2313         if authorization fails, user specified IP address
2314         is invalid or server cannot be added because
2315         server group is full
2316 </comment>
2317 </entry>
2318 <entry id="server_group">
2319     <internal token="text"/>
2320     <external opt="required" type="char *"/>
2321     <comment>server group name</comment>
2322 </entry>
2323 <entry id="server_minport">
2324     <internal token="iport"/>
2325     <external opt="optional" type="uint16_t" />
2326     <comment>server's minimum value in port range - empty
2327         means default value (see man page)
2328 </comment>
2329 </entry>
2330 <entry id="server_maxport">
2331     <internal token="iport"/>
2332     <external opt="optional" type="uint16_t" />
2333     <comment>server's maximum value in port range - empty
2334         means default value(see man page)
2335 </comment>
2336 </entry>
2337 <entry id="return">
2338     <internal token="return"/>
2339     <external opt="none"/>
2340 </entry>
2341 </event>
2342
2343 <event id="AUE_ilb_disable_server" header="0" idNo="127" omit="JNI">
2344     <title>Disable server to Integrated Loadbalancer</title>
2345     <program>/usr/sbin/ilbadm</program>
2346     <see>ilbadm(1m)</see>
2347     <entry id="subject">
2348         <internal token="subject"/>
2349         <external opt="none"/>
2350     </entry>
2351     <entry id="auth_used">
2352         <internal token="uauth"/>
2353         <external opt="required" type="char *"/>
2354         <comment>authorization used</comment>
2355     </entry>
2356     <entry id="server_id">
2357         <internal token="text"/>
2358         <external opt="required" type="char *"/>
2359         <comment>serverid</comment>
2360 </entry>
2361 <entry id="server_ipaddress_type, server_ipaddress">
2362     <internal token="in_remote"/>
2363     <external opt="optional" type="int32_t, uint32_t[4]"/>
2364     <comment>IPaddr corresponding to the serverid - empty
2365         if authorization fails, or user specified serverid
2366         is nonexistent
2367 </comment>
2368 </entry>
2369 <entry id="return">
2370     <internal token="return"/>
2371     <external opt="none"/>

```

```

2372     </entry>
2373 </event>

2375 <event id="AUE_ilb_enable_server" header="0" idNo="128" omit="JNI">
2376   <title>Enable server to Integrated Loadbalancer</title>
2377   <program>/usr/sbin/ilbadm</program>
2378   <see>ilbadm(1m)</see>
2379   <entry id="subject">
2380     <internal token="subject"/>
2381     <external opt="none"/>
2382   </entry>
2383   <entry id="auth_used">
2384     <internal token="uauth"/>
2385     <external opt="required" type="char *"/>
2386     <comment>authorization used</comment>
2387   </entry>
2388   <entry id="server_id">
2389     <internal token="text"/>
2390     <external opt="required" type="char *"/>
2391     <comment>serverid</comment>
2392   </entry>
2393   <entry id="server_ipaddress_type,server_ipaddress">
2394     <internal token="in_remote"/>
2395     <external opt="optional" type="int32_t,uint32_t[4]"/>
2396     <comment>IPAddr corresponding to the serverid - empty
2397       if authorization fails, or user specified serverid
2398       is nonexistent
2399     </comment>
2400   </entry>
2401   <entry id="return">
2402     <internal token="return"/>
2403     <external opt="none"/>
2404   </entry>
2405 </event>

2407 <event id="AUE_ilb_remove_server" header="0" idNo="129" omit="JNI">
2408   <title>Remove server from Integrated Loadbalancer</title>
2409   <program>/usr/sbin/ilbadm</program>
2410   <see>ilbadm(1m)</see>
2411   <entry id="subject">
2412     <internal token="subject"/>
2413     <external opt="none"/>
2414   </entry>
2415   <entry id="auth_used">
2416     <internal token="uauth"/>
2417     <external opt="required" type="char *"/>
2418     <comment>authorization used</comment>
2419   </entry>
2420   <entry id="server_id">
2421     <internal token="text"/>
2422     <external opt="required" type="char *"/>
2423     <comment>serverid</comment>
2424   </entry>
2425   <entry id="server_group">
2426     <internal token="text"/>
2427     <external opt="required" type="char *"/>
2428     <comment>server group name</comment>
2429   </entry>
2430   <entry id="server_ipaddress_type,server_ipaddress">
2431     <internal token="in_remote"/>
2432     <external opt="optional" type="int32_t,uint32_t[4]"/>
2433     <comment>IPAddr corresponding to serverid - empty
2434       if authorization fails or user specified serverid
2435       serverid is nonexistent
2436     </comment>
2437   </entry>

```

```

2438     <entry id="return">
2439       <internal token="return"/>
2440       <external opt="none"/>
2441     </entry>
2442 </event>

2444 <event id="AUE_ilb_create_servergroup" header="0" idNo="130" omit="JNI">
2445   <title>Create server group for Integrated Loadbalancer</title>
2446   <program>/usr/sbin/ilbadm</program>
2447   <see>ilbadm(1m)</see>
2448   <entry id="subject">
2449     <internal token="subject"/>
2450     <external opt="none"/>
2451   </entry>
2452   <entry id="auth_used">
2453     <internal token="uauth"/>
2454     <external opt="required" type="char *"/>
2455     <comment>authorization used</comment>
2456   </entry>
2457   <entry id="server_group">
2458     <internal token="text"/>
2459     <external opt="required" type="char *"/>
2460     <comment>server group name</comment>
2461   </entry>
2462   <entry id="return">
2463     <internal token="return"/>
2464     <external opt="none"/>
2465   </entry>
2466 </event>

2468 <event id="AUE_ilb_delete_servergroup" header="0" idNo="131" omit="JNI">
2469   <title>Delete server group from Integrated Loadbalancer</title>
2470   <program>/usr/sbin/ilbadm</program>
2471   <see>ilbadm(1m)</see>
2472   <entry id="subject">
2473     <internal token="subject"/>
2474     <external opt="none"/>
2475   </entry>
2476   <entry id="auth_used">
2477     <internal token="uauth"/>
2478     <external opt="required" type="char *"/>
2479     <comment>authorization used</comment>
2480   </entry>
2481   <entry id="server_group">
2482     <internal token="text"/>
2483     <external opt="required" type="char *"/>
2484     <comment>server group name</comment>
2485   </entry>
2486   <entry id="return">
2487     <internal token="return"/>
2488     <external opt="none"/>
2489   </entry>
2490 </event>

2492 <event id="AUE_nwam_enable" header="0" idNo="132" omit="JNI">
2493   <entry id="subject">
2494     <internal token="subject"/>
2495     <external opt="none"/>
2496   </entry>
2497   <entry id="profile_type">
2498     <internal token="text"/>
2499     <external opt="required" type="char *"/>
2500     <comment>Type of profile being enabled</comment>
2501   </entry>
2502   <entry id="profile_name">
2503     <internal token="text"/>

```



```

2504         <external opt="required" type="char *"/>
2505         <comment>Name of profile being enabled</comment>
2506     </entry>
2507     <entry id="return">
2508         <internal token="return"/>
2509         <external opt="none"/>
2510     </entry>
2511 </event>

2513 <event id="AUE_nwam_disable" header="0" idNo="133" omit="JNI">
2514     <entry id="subject">
2515         <internal token="subject"/>
2516         <external opt="none"/>
2517     </entry>
2518     <entry id="profile_type">
2519         <internal token="text"/>
2520         <external opt="required" type="char *"/>
2521         <comment>Type of profile being disabled</comment>
2522     </entry>
2523     <entry id="profile_name">
2524         <internal token="text"/>
2525         <external opt="required" type="char *"/>
2526         <comment>Name of profile being disabled</comment>
2527     </entry>
2528     <entry id="return">
2529         <internal token="return"/>
2530         <external opt="none"/>
2531     </entry>
2532 </event>

2534 <event id="AUE_netcfg_update" header="0" idNo="134" omit="JNI">
2535     <entry id="subject">
2536         <internal token="subject"/>
2537         <external opt="none"/>
2538     </entry>
2539     <entry id="parent_file">
2540         <internal token="text"/>
2541         <external opt="required" type="char *"/>
2542         <comment>Back-end data file being updated</comment>
2543     </entry>
2544     <entry id="object_name">
2545         <internal token="text"/>
2546         <external opt="required" type="char *"/>
2547         <comment>Name of object being updated</comment>
2548     </entry>
2549     <entry id="return">
2550         <internal token="return"/>
2551         <external opt="none"/>
2552     </entry>
2553 </event>

2555 <event id="AUE_netcfg_remove" header="0" idNo="135" omit="JNI">
2556     <entry id="subject">
2557         <internal token="subject"/>
2558         <external opt="none"/>
2559     </entry>
2560     <entry id="parent_file">
2561         <internal token="text"/>
2562         <external opt="required" type="char *"/>
2563         <comment>Back-end data file being modified</comment>
2564     </entry>
2565     <entry id="object_name">
2566         <internal token="text"/>
2567         <external opt="required" type="char *"/>
2568         <comment>Name of object being removed</comment>
2569     </entry>

```

```

2570     <entry id="return">
2571         <internal token="return"/>
2572         <external opt="none"/>
2573     </entry>
2574 </event>
2575 <event id="AUE_uadmin_intrd" instance_of="AUE_uadmin_generic_fcn">
2576     header="0" idNo="136" omit="JNI">
2577         <title>uadmin intrd</title>
2578         <program>/sbin/uadmin</program>
2579         <program>/usr/sbin/uadmin</program>
2580         <see>uadmin(1M)</see>
2581     </event>
2582 #endif /* ! codereview */

2584 <!-- add new events here with the next higher idNo -->
2585 <!-- Highest idNo is 135, so next is 136, then fix this comment -->
2586 <!-- end of C Only events -->

2588 <!--
2589     token definitions are partially implemented. All they do for now
2590     is create a list of defined token names. In the future they may
2591     become a way of describing token structure.
2592 -->

2594 <token id="acl">
2595 </token>
2596 <token id="arbitrary">
2597 </token>
2598 <token id="arg">
2599 </token>
2600 <token id="attr">
2601 </token>
2602 <token id="command">
2603 </token>
2604 <token id="command_alt">
2605 </token>
2606 <token id="date">
2607 </token>
2608 <token id="exec_args">
2609 </token>
2610 <token id="exec_env">
2611 </token>
2612 <token id="exit">
2613 </token>
2614 <token id="file">
2615 </token>
2616 <token id="fmri">
2617 </token>
2618 <token id="groups">
2619 </token>
2620 <!--
2621     the iport token take a single argument of type uint16_t
2622     if there are any other tokens following it that have arguments
2623     the last of the iport tokens in the event description must
2624     be followed by a dummy iport token that is optional.
2625     This is to ensure proper structure alignment across all
2626     compilers and architectures.
2627 -->
2628 <token id="iport">
2629 </token>
2630 <!-- pseudo token; in_addr and in_port of peer -->
2631 <token id="in_peer">
2632 </token>
2633 <!-- pseudo token; specified in_addr -->
2634 <token id="in_remote">
2635 </token>

```

```

2636 <token id="ipc">
2637 </token>
2638 <token id="ipc_perm">
2639 </token>
2640 <token id="label">
2641 </token>
2642 <token id="newgroups">
2643 </token>
2644 <token id="opaque">
2645 </token>
2646 <token id="path">
2647 </token>
2648 <!-- pseudo token; path list generates 0 or more path tokens -->
2649 <token id="path_list">
2650 </token>
2651 <!--
2652     privilege token is implemented as one of the pseudo tokens
2653     priv_limit, priv_effective, or priv_inherit
2654 -->
2655 <token id="privilege">
2656 </token>
2657 -->
2658 <token id="priv_effective">
2659 </token>
2660 <token id="priv_inherit">
2661 </token>
2662 <token id="priv_limit">
2663 </token>
2664 <token id="process">
2665 </token>
2666 <token id="return">
2667 </token>
2668 <token id="seq">
2669 </token>
2670 <token id="socket">
2671 </token>
2672 <token id="socket_inet">
2673 </token>
2674 <token id="subject">
2675 </token>
2676 <token id="text">
2677 </token>
2678 <token id="tid">
2679 </token>
2680 <token id="uauth">
2681 </token>
2682 <token id="user">
2683 </token>
2684 <token id="zonename">
2685 </token>
2686
2687 <!--
2688     error value list for return values with success/fail code of fail.
2689     These values start at 1000 so praudit can tell the difference
2690     between the libbsm/common/audit_*.c broken error values and
2691     the new adt_ error value list. It is public so that praudit
2692     can find it.
2693 -->
2694     praudit outputs "failure" %s" for these strings, so there is
2695     no need to use words such as "failed" in the message.
2696
2697     ** Add to the end only to maintain validity across versions of
2698     the audit log. **
2699 -->
2701 <msg_list id="fail_value" header="0" start="1000" public="true">

```

```

2702 <msg id="PW_ATTR">Attribute update</msg>
2703 <msg id="PW">Password update</msg>
2704 <msg id="USERNAME">bad username</msg>
2705 <msg id="AUTH">authorization failed</msg>
2706 <msg id="UID">bad uid</msg>
2707 <msg id="UNKNOWN">unknown failure</msg>
2708 <msg id="EXPIRED">password expired</msg>
2709 <msg id="ACCOUNT_LOCKED">Account is locked</msg>
2710 <msg id="BAD_DIALUP">Bad dial up</msg>
2711 <msg id="BAD_ID">Invalid ID</msg>
2712 <msg id="BAD_PW">Invalid password</msg>
2713 <msg id="CONSOLE">Not on console</msg>
2714 <msg id="MAX_TRIES">Too many failed attempts</msg>
2715 <msg id="PROTOCOL_FAILURE">Protocol failure</msg>
2716 <msg id="EXCLUDED_USER">Excluded user</msg>
2717 <msg id="ANON_USER">No anonymous</msg>
2718 <msg id="BAD_CMD">Invalid command</msg>
2719 <msg id="BAD_TTY">Standard input not a tty line</msg>
2720 <msg id="PROGRAM">Program failure</msg>
2721 <msg id="CHDIR_FAILED">chdir to home directory</msg>
2722 <msg id="INPUT_OVERFLOW">Input line too long</msg>
2723 <msg id="DEVICE_PERM">login device override</msg>
2724 <msg id="AUTH_BYPASS">authorization bypass</msg>
2725 <msg id="LOGIN_DISABLED">login disabled</msg>
2726 </msg_list>
2727
2728 <!--
2729     The following empty list is used for PAM errors; the "start"
2730     value is used by praudit to know to use the PAM infrastructure
2731     for generating error strings
2732 -->
2733 <msg_list id="fail_pam" header="0" start="2000" public="true">
2734 </msg_list>
2735
2736 <!--
2737     This is still in use by SMC. See AUE_generic_login. When
2738     either SMC is fixed to stop using this, or SMC goes away.
2739     REMOVE this stuff and the corresponding AUE_generic_login
2740     message field.
2741 -->
2742     Message list for the various authentication events, such
2743     as AUE_login and AUE_admin_authenticate. Add new entries
2744     at the end. The order of msg_list entries and the order
2745     of msg entries both affect the names in adt.h and the value
2746     of the associated enumerated types.
2747
2748     Each of these messages except NO_MSG is also in the failure_attribute
2749     list; the difference is that the messages below use a text token
2750     in the audit record, while the failure_attribute messages are
2751     associated with the return value of the return token.
2752
2753     This list is deprecated; please don't use text tokens for error
2754     messages.
2755 -->
2756
2757 <msg_list id="login_text" header="0" deprecated="true">
2758 <msg id="NO_MSG"></msg>
2759 <msg id="ACCOUNT_LOCKED">Account is locked</msg>
2760 <msg id="BAD_DIALUP">Bad dial up</msg>
2761 <msg id="BAD_ID">Invalid ID</msg>
2762 <msg id="BAD_PW">Invalid password</msg>
2763 <msg id="CONSOLE">Not on console</msg>
2764 <msg id="MAX_TRIES">Too many failed attempts</msg>
2765 <msg id="PROTOCOL_FAILURE">Protocol failure</msg>
2766 <msg id="EXCLUDED_USER">Excluded user</msg>
2767 <msg id="ANON_USER">No anonymous</msg>

```

```

2768 </msg_list>
2770 <!-- msg list for uadmin(1m) fcn argument (next action, see uadmin(2)) -->
2771 <msg_list id="uadmin_fcn" header="0" start="3000" public="true">
2772 <msg id="AD_HALT">Halt the processor(s)</msg>
2773 <msg id="AD_POWEROFF">Halt the processor(s) and turn off the power</msg>
2774 <msg id="AD_BOOT">Reboot the system using the kernel file</msg>
2775 <msg id="AD_IBOOT">Interactive reboot</msg>
2776 <msg id="AD_SUSPEND_TO_DISK">Save the system state to the state file</ms
2777 <msg id="AD_CHECK_SUSPEND_TO_DISK">Check if system supports suspend to d
2778 <msg id="AD_FORCE">Force suspend to disk even when threads of user
2779 applications are not suspendable</msg>
2780 <msg id="AD_SUSPEND_TO_RAM">Save the system state to memory</msg>
2781 <msg id="AD_CHECK_SUSPEND_TO_RAM">Check if system supports suspend to me
2782 <msg id="AD_SBOOT">Single-user reboot</msg>
2783 <msg id="AD_SIBOOT">Single-user interactive reboot</msg>
2784 <msg id="AD_NOSYNC">Do not sync filesystems on next A_DUMP</msg>
2785 <msg id="AD_FASTREBOOT">Reboot bypassing BIOS and boot loader</msg>
2786 <msg id="AD_FASTREBOOT_DRYRUN">Check if system supports reboot bypassing
2787 <msg id="AD_UPDATE_BOOT_CONFIG">Update boot configuration parameters</ms
2788 <msg id="AD_REUSEINIT">Prepare for AD_REUSABLE</msg>
2789 <msg id="AD_REUSABLE">Create reusable statefile</msg>
2790 <msg id="AD_REUSEFINI">Revert to normal CPR mode (not reusable)</msg>
2791 <msg id="AD_FTRACE_START">ftrace start</msg>
2792 <msg id="AD_FTRACE_STOP">ftrace stop</msg>
2793 </msg_list>
2795 <!--
2796 msg list for TPM errors that will be reported by tcstd(8).
2797 This list must match the order of the TPM_E_* error codes defined
2798 in /usr/include/tss/tpm_error.h (SUNWtss package)
2799 -->
2800 <msg_list id="tpm_e" header="0" start="4000" public="true">
2801 <msg id="AUTHFAIL">Authentication failed</msg>
2802 <msg id="BADINDEX">The index to a PCR, DIR or other register is incorrec
2803 <msg id="BAD_PARAMETER">One or more parameter is bad</msg>
2804 <msg id="AUDITFAILURE">auditing of the operation failed.</msg>
2805 <msg id="CLEAR_DISABLED">clear operations now physical access</msg>
2806 <msg id="DEACTIVATED">The TPM is deactivated</msg>
2807 <msg id="DISABLED">The TPM is disabled</msg>
2808 <msg id="DISABLED_CMD">The target command has been disabled</msg>
2809 <msg id="FAIL">The operation failed</msg>
2810 <msg id="BAD_ORDINAL">The ordinal was unknown or inconsistent</msg>
2811 <msg id="INSTALL_DISABLED">The ability to install an owner is disabled</
2812 <msg id="INVALID_KEYHANDLE">The key handle can not be interpreted</msg>
2813 <msg id="KEYNOTFOUND">The key handle points to an invalid key</msg>
2814 <msg id="INAPPROPRIATE_ENC">Unacceptable encryption scheme</msg>
2815 <msg id="MIGRATEFAIL">Migration authorization failed</msg>
2816 <msg id="INVALID_PCR_INFO">PCR information could not be interpreted</msg>
2817 <msg id="NOSPACE">No room to load key.</msg>
2818 <msg id="NOSRK">There is no SRK set</msg>
2819 <msg id="NOTSEALED_BLOB">An encrypted blob is invalid or was
2820 not created by this TPM</msg>
2821 <msg id="OWNER_SET">There is already an Owner </msg>
2822 <msg id="RESOURCES">The TPM has insufficient internal resources</msg>
2823 <msg id="SHORTRANDOM">A random string was too short</msg>
2824 <msg id="SIZE">The TPM does not have the space to perform the operation.
2825 <msg id="WRONGPCRVAL">The named PCR value does not match the current PCR
2826 <msg id="BAD_PARAM_SIZE">The paramSize argument has the incorrect value
2827 <msg id="SHA_THREAD">There is no existing SHA-1 thread.</msg>
2828 <msg id="SHA_ERROR">SHA-1 thread encountered an error.</msg>
2829 <msg id="FAILEDSELFTEST">Self-test has failed and the TPM has shutdown.<
2830 <msg id="AUTH2FAIL">The auth for the second key failed authorization</ms
2831 <msg id="BADTAG">The tag value sent to for a command is invalid</msg>
2832 <msg id="IOERROR">An IO error occurred transmitting information to the T
2833 <msg id="ENCRYPT_ERROR">The encryption process had a problem.</msg>

```

```

2834 <msg id="DECRYPT_ERROR">The decryption process did not complete.</msg>
2835 <msg id="INVALID_AUTHHANDLE">An invalid handle was used.</msg>
2836 <msg id="NO_ENDORSEMENT">The TPM does not have an EK installed</msg>
2837 <msg id="INVALID_KEYUSAGE">The usage of a key is not allowed</msg>
2838 <msg id="WRONG_ENTITYTYPE">The submitted entity type is not allowed</msg>
2839 <msg id="INVALID_POSTINIT">The command was received in the wrong sequenc
2840 <msg id="INAPPROPRIATE_SIG">Signed data cannot include additional DER in
2841 <msg id="BAD_KEY_PROPERTY">The key properties are not supported by this
2842 <msg id="BAD_MIGRATION">The migration properties of this key are incorre
2843 <msg id="BAD_SCHEME">Incorrect signature or encryption scheme</msg>
2844 <msg id="BAD_DATASIZE">The size of the data parameter is bad</msg>
2845 <msg id="BAD_MODE">A mode parameter is bad</msg>
2846 <msg id="BAD_PRESENCE">physicalPresence or physicalPresenceLock bits hav
2847 <msg id="BAD_VERSION">The TPM cannot perform this version of the capabil
2848 <msg id="NO_WRAP_TRANSPORT">The TPM does not allow for wrapped transport
2849 <msg id="AUDITFAIL_UNSUCCESSFUL">TPM audit construction failed for faile
2850 <msg id="AUDITFAIL_SUCCESSFUL">TPM audit construction failed for success
2851 <msg id="NOTRESETTABLE">PCR register does not have the resettable attribu
2852 <msg id="NOTLOCAL">PCR register requires locality</msg>
2853 <msg id="BAD_TYPE">Make identity blob not properly typed</msg>
2854 <msg id="INVALID_RESOURCE">Resource type does not match actual resource<
2855 <msg id="NOTFIPS">Command only available when TPM is in FIPS mode</msg>
2856 <msg id="INVALID_FAMILY">The command is attempting to use an invalid fam
2857 <msg id="NO_NV_PERMISSION">The permission to manipulate the NV storage i
2858 <msg id="REQUIRES_SIGN">The operation requires a signed command</msg>
2859 <msg id="KEY_NOTSUPPORTED">Wrong operation to load an NV key</msg>
2860 <msg id="AUTH_CONFLICT">NV LoadKey blob requires both owner and blob aut
2861 <msg id="AREA_LOCKED">The NV area is locked and not writable</msg>
2862 <msg id="BAD_LOCALITY">The locality is incorrect for the attempted opera
2863 <msg id="READ_ONLY">The NV area is read only and can't be written to</ms
2864 <msg id="PER_NOWRITE">There is no protection on the write to the NV area
2865 <msg id="FAMILYCOUNT">The family count value does not match</msg>
2866 <msg id="WRITE_LOCKED">The NV area has already been written to</msg>
2867 <msg id="BAD_ATTRIBUTES">The NV area attributes conflict</msg>
2868 <msg id="INVALID_STRUCTURE">The tag and version are invalid or inconsist
2869 <msg id="KEY_OWNER_CONTROL">The key evicted by the TPM Owner.</msg>
2870 <msg id="BAD_COUNTER">The counter handle is incorrect</msg>
2871 <msg id="NOT_FULLWRITE">The write is not a complete write of the area</m
2872 <msg id="CONTEXT_GAP">The gap between saved context counts is too large<
2873 <msg id="MAXNVWRITES">Max number of NV writes without owner has been exc
2874 <msg id="NOOPERATOR">No operator AuthData value is set</msg>
2875 <msg id="RESOURCEMISSING">The resource pointed to by context is not load
2876 <msg id="DELEGATE_LOCK">The delegate administration is locked</msg>
2877 <msg id="DELEGATE_FAMILY">Attempt to manage a family other than the dele
2878 <msg id="DELEGATE_ADMIN">Delegation table management not enabled</msg>
2879 <msg id="TRANSPORT_NOTEXCLUSIVE">Command executed outside of exclusive t
2880 <msg id="OWNER_CONTROL">Attempt to context save a owner evict controlled
2881 <msg id="DAA_RESOURCES">DAA command has no resources available to execut
2882 <msg id="DAA_INPUT_DATA0">The consistency check on DAA parameter inputDa
2883 <msg id="DAA_INPUT_DATA1">The consistency check on DAA parameter inputDa
2884 <msg id="DAA_ISSUER_SETTINGS">The consistency check on DAA issuerSetting
2885 <msg id="DAA_TPM_SETTINGS">The consistency check on DAA tpmSpecific has
2886 <msg id="DAA_STAGE">Atomic process indicated by DAA command is not the e
2887 <msg id="DAA_ISSUER_VALIDITY">Inconsistent issuer validity</msg>
2888 <msg id="DAA_WRONG_W">The consistency check on w has failed.</msg>
2889 <msg id="BAD_HANDLE">The handle is incorrect</msg>
2890 <msg id="BAD_DELEGATE">Delegation is not correct</msg>
2891 <msg id="BADCONTEXT">The context blob is invalid</msg>
2892 <msg id="TOOMANYCONTEXTS">Too many contexts held by the TPM</msg>
2893 <msg id="MA_TICKET_SIGNATURE">Migration authority signature validation f
2894 <msg id="MA_DESTINATION">Migration destination not authenticated</msg>
2895 <msg id="MA_SOURCE">Migration source incorrect</msg>
2896 <msg id="MA_AUTHORITY">Incorrect migration authority</msg>
2897 <msg id="PERMANENTEK">Attempt to revoke the EK and the EK is not revocab
2898 <msg id="BAD_SIGNATURE">Bad signature of CMK ticket</msg>
2899 <msg id="NOCONTEXTSPACE">There is no room in the context list for additi

```

```
2900     <msg id="RETRY">The TPM is too busy to respond to the command immediatel
2901     <msg id="NEEDS_SELFTEST">SelfTestFull has not been run</msg>
2902     <msg id="DOING_SELFTEST">The TPM is currently executing a full selftest<
2903     <msg id="DEFEND_LOCK_RUNNING">TPM is defending against dictionary attack
2904     <msg id="NO_MSG"></msg>
2905     <!-- End TPM failure codes -->
2906 </msg_list>
2907 </specification>
```

new/usr/src/pkg/manifests/SUNWcs.mf

1

89496 Sat Sep 14 15:00:28 2013

new/usr/src/pkg/manifests/SUNWcs.mf

XXX Rearchitect and replace interrupt distribution

```
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright 2012 Nexenta Systems, Inc. All rights reserved.
25 #
26 #
27 <include SUNWcs.man1.inc>
28 <include SUNWcs.man1m.inc>
29 <include SUNWcs.man4.inc>
30 <include SUNWcs.man5.inc>
31 <include SUNWcs.man7d.inc>
32 <include SUNWcs.man7fs.inc>
33 set name=pkg.fmri value=pkg:/SUNWcs@$(PKGVERS)
34 set name=pkg.description \
35     value="core software for a specific instruction-set architecture"
36 set name=pkg.summary value="Core Solaris"
37 set name=info.classification value=org.opensolaris.category.2008:System/Core
38 set name=variant.arch value=$(ARCH)
39 dir path=dev group=sys
40 dir path=etc group=sys
41 dir path=etc/cron.d group=sys
42 dir path=etc/crypto group=sys
43 dir path=etc/crypto/certs group=sys
44 dir path=etc/crypto/crls group=sys
45 dir path=etc/default group=sys
46 dir path=etc/dev group=sys
47 dir path=etc/devices group=sys
48 dir path=etc/dfs group=sys
49 dir path=etc/dhcp group=sys
50 dir path=etc/fs group=sys
51 dir path=etc/fs/dev group=sys
52 dir path=etc/fs/hsfs group=sys
53 dir path=etc/fs/ufs group=sys
54 dir path=etc/ftpd group=sys
55 dir path=etc/inet group=sys
56 dir path=etc/init.d group=sys
57 dir path=etc/lib group=sys
58 dir path=etc/logadm.d group=sys
59 dir path=etc/mail group=mail
60 dir path=etc/net group=sys
61 dir path=etc/net/ticots group=sys
```

new/usr/src/pkg/manifests/SUNWcs.mf

2

```
62 dir path=etc/net/ticots group=sys
63 dir path=etc/net/ticotsord group=sys
64 dir path=etc/opt group=sys
65 dir path=etc/rc0.d group=sys
66 dir path=etc/rc1.d group=sys
67 dir path=etc/rc2.d group=sys
68 dir path=etc/rc3.d group=sys
69 dir path=etc/rcS.d group=sys
70 dir path=etc/rpcsec group=sys
71 dir path=etc/saf
72 dir path=etc/saf/zsmon group=sys
73 dir path=etc/sasl group=sys
74 dir path=etc/security group=sys
75 dir path=etc/security/audit group=sys
76 dir path=etc/security/audit/localhost group=sys
77 dir path=etc/security/auth_attr.d group=sys
78 dir path=etc/security/dev group=sys
79 dir path=etc/security/exec_attr.d group=sys
80 dir path=etc/security/lib group=sys
81 dir path=etc/security/prof_attr.d group=sys
82 dir path=etc/skel group=sys
83 dir path=etc/svc group=sys
84 dir path=etc/svc/profile group=sys
85 dir path=etc/svc/profile/site group=sys
86 dir path=etc/svc/volatile group=sys
87 dir path=etc/sysevent group=sys
88 dir path=etc/sysevent/config group=sys
89 dir path=etc/tm group=sys
90 dir path=etc/user_attr.d group=sys
91 dir path=export group=sys
92 dir path=home group=root mode=0555
93 dir path=lib
94 dir path=lib/crypto
95 dir path=lib/inet
96 dir path=lib/svc
97 dir path=lib/svc/bin
98 dir path=lib/svc/capture
99 dir path=lib/svc/manifest group=sys
100 dir path=lib/svc/manifest/application group=sys
101 dir path=lib/svc/manifest/application/management group=sys
102 dir path=lib/svc/manifest/application/security group=sys
103 dir path=lib/svc/manifest/device group=sys
104 dir path=lib/svc/manifest/milestone group=sys
105 dir path=lib/svc/manifest/network group=sys
106 dir path=lib/svc/manifest/network/dns group=sys
107 dir path=lib/svc/manifest/network/ipsec group=sys
108 dir path=lib/svc/manifest/network/ldap group=sys
109 dir path=lib/svc/manifest/network/routing group=sys
110 dir path=lib/svc/manifest/network/rpc group=sys
111 dir path=lib/svc/manifest/network/shares group=sys
112 dir path=lib/svc/manifest/network/ssl group=sys
113 dir path=lib/svc/manifest/platform group=sys
114 $(sparc_ONLY)dir path=lib/svc/manifest/platform/sun4u group=sys
115 dir path=lib/svc/manifest/site group=sys
116 dir path=lib/svc/manifest/system group=sys
117 dir path=lib/svc/manifest/system/device group=sys
118 dir path=lib/svc/manifest/system/filesystem group=sys
119 dir path=lib/svc/manifest/system/security group=sys
120 dir path=lib/svc/manifest/system/svc group=sys
121 dir path=lib/svc/method
122 dir path=lib/svc/monitor
123 dir path=lib/svc/seed
124 dir path=lib/svc/share
125 dir path=mnt group=sys
126 dir path=opt group=sys
127 dir path=proc group=root mode=0555
```

new/usr/src/pkg/manifests/SUNWcs.mf

3

```
128 dir path=root group=root mode=0700
129 dir path=sbin group=sys
130 dir path=system group=root
131 dir path=system/contract group=root mode=0555
132 dir path=system/object group=root mode=0555
133 dir path=tmp group=sys mode=1777
134 dir path=usr group=sys
135 dir path=usr/bin
136 dir path=usr/bin/$(ARCH32)
137 dir path=usr/bin/$(ARCH64)
138 dir path=usr/ccs
139 dir path=usr/ccs/bin
140 dir path=usr/demo
141 dir path=usr/games
142 dir path=usr/has
143 dir path=usr/has/bin
144 dir path=usr/has/lib
145 dir path=usr/has/man
146 dir path=usr/has/man/man1has
147 dir path=usr/kernel group=sys
148 dir path=usr/kernel/drv group=sys
149 dir path=usr/kernel/drv/$(ARCH64) group=sys
150 dir path=usr/kernel/exec group=sys
151 dir path=usr/kernel/exec/$(ARCH64) group=sys
152 dir path=usr/kernel/fs group=sys
153 dir path=usr/kernel/fs/$(ARCH64) group=sys
154 dir path=usr/kernel/pcbe group=sys
155 dir path=usr/kernel/pcbe/$(ARCH64) group=sys
156 dir path=usr/kernel/sched group=sys
157 dir path=usr/kernel/sched/$(ARCH64) group=sys
158 dir path=usr/kernel/strmod group=sys
159 dir path=usr/kernel/strmod/$(ARCH64) group=sys
160 dir path=usr/kernel/sys group=sys
161 dir path=usr/kernel/sys/$(ARCH64) group=sys
162 dir path=usr/kvm
163 dir path=usr/lib
164 dir path=usr/lib/$(ARCH64)
165 dir path=usr/lib/audit
166 dir path=usr/lib/class
167 dir path=usr/lib/class/FX
168 dir path=usr/lib/class/IA
169 dir path=usr/lib/class/RT
170 dir path=usr/lib/class/SDC
171 dir path=usr/lib/class/TS
172 dir path=usr/lib/crypto
173 dir path=usr/lib/devfsadm group=sys
174 dir path=usr/lib/devfsadm/linkmod group=sys
175 dir path=usr/lib/fs group=sys
176 dir path=usr/lib/fs/autofs group=sys
177 dir path=usr/lib/fs/autofs/$(ARCH64) group=sys
178 dir path=usr/lib/fs/cacheefs group=sys
179 dir path=usr/lib/fs/ctfs group=sys
180 dir path=usr/lib/fs/dev group=sys
181 dir path=usr/lib/fs/fd group=sys
182 dir path=usr/lib/fs/hsfs group=sys
183 dir path=usr/lib/fs/lofs group=sys
184 dir path=usr/lib/fs/mntfs group=sys
185 dir path=usr/lib/fs/nfs group=sys
186 dir path=usr/lib/fs/nfs/$(ARCH64) group=sys
187 dir path=usr/lib/fs/objfs group=sys
188 dir path=usr/lib/fs/proc group=sys
189 dir path=usr/lib/fs/sharefs group=sys
190 dir path=usr/lib/fs/tmpfs group=sys
191 dir path=usr/lib/fs/ufs group=sys
192 dir path=usr/lib/help
193 dir path=usr/lib/help/auths
```

new/usr/src/pkg/manifests/SUNWcs.mf

4

```
194 dir path=usr/lib/help/auths/locale
195 dir path=usr/lib/help/auths/locale/C
196 dir path=usr/lib/help/profiles
197 dir path=usr/lib/help/profiles/locale
198 dir path=usr/lib/help/profiles/locale/C
199 dir path=usr/lib/iconv
200 dir path=usr/lib/inet
201 dir path=usr/lib/inet/$(ARCH32)
202 dir path=usr/lib/inet/$(ARCH64)
203 dir path=usr/lib/inet/dhcp
204 dir path=usr/lib/inet/dhcp/nsu
205 dir path=usr/lib/inet/dhcp/svc
206 dir path=usr/lib/locale
207 dir path=usr/lib/locale/C
208 dir path=usr/lib/locale/C/LC_COLLATE
209 dir path=usr/lib/locale/C/LC_CTYPE
210 dir path=usr/lib/locale/C/LC_MESSAGES
211 dir path=usr/lib/locale/C/LC_MONETARY
212 dir path=usr/lib/locale/C/LC_NUMERIC
213 dir path=usr/lib/locale/C/LC_TIME
214 dir path=usr/lib/netsvc group=sys
215 dir path=usr/lib/pci
216 dir path=usr/lib/rcm
217 dir path=usr/lib/rcm/modules
218 dir path=usr/lib/rcm/scripts
219 dir path=usr/lib/reparse
220 dir path=usr/lib/saf
221 dir path=usr/lib/secure
222 dir path=usr/lib/secure/$(ARCH64)
223 dir path=usr/lib/security
224 dir path=usr/lib/sysevent
225 dir path=usr/lib/sysevent/modules
226 dir path=usr/net group=sys
227 dir path=usr/net/nls group=sys
228 dir path=usr/net/servers group=sys
229 dir path=usr/old
230 dir path=usr/platform group=sys
231 dir path=usr/sadm
232 dir path=usr/sadm/bin
233 dir path=usr/sadm/install
234 dir path=usr/sadm/install/scripts
235 dir path=usr/sbin
236 $(i386_ONLY)dir path=usr/sbin/$(ARCH32)
237 dir path=usr/sbin/$(ARCH64)
238 dir path=usr/share
239 dir path=usr/share/doc group=other
240 dir path=usr/share/doc/ksh
241 dir path=usr/share/doc/ksh/images
242 dir path=usr/share/doc/ksh/images/callouts
243 dir path=usr/share/lib
244 dir path=usr/share/lib/mailx
245 dir path=usr/share/lib/pub
246 dir path=usr/share/lib/tabset
247 dir path=usr/share/lib/xml group=sys
248 dir path=usr/share/lib/xml/dtd group=sys
249 dir path=usr/share/lib/xml/style group=sys
250 dir path=usr/share/man
251 dir path=usr/share/man/man1
252 dir path=usr/share/man/man1m
253 dir path=usr/share/man/man4
254 dir path=usr/share/man/man5
255 dir path=usr/share/man/man7d
256 dir path=usr/share/man/man7fs
257 dir path=usr/share/src group=sys
258 dir path=var group=sys
259 dir path=var/adm group=sys mode=0775
```

```

260 dir path=var/adm/exacct group=adm owner=adm
261 dir path=var/adm/log group=adm owner=adm
262 dir path=var/adm/streams group=sys
263 dir path=var/audit group=sys
264 dir path=var/cores group=sys
265 dir path=var/cron group=sys
266 dir path=var/games
267 dir path=var/ldap group=daemon owner=daemon
268 dir path=var/inet group=sys
269 dir path=var/ld
270 dir path=var/ld/$(ARCH64)
271 dir path=var/log group=sys
272 dir path=var/logadm
273 dir path=var/mail group=mail mode=1777
274 dir path=var/mail/:saved group=mail mode=0775
275 dir path=var/news
276 dir path=var/opt group=sys
277 dir path=var/preserve mode=1777
278 dir path=var/run group=sys
279 dir path=var/sadm group=sys
280 dir path=var/sadm/system group=sys
281 dir path=var/sadm/system/admin group=sys
282 dir path=var/saf
283 dir path=var/saf/zsmon group=sys
284 dir path=var/spool
285 dir path=var/spool/cron group=sys
286 dir path=var/spool/cron/atjobs group=sys
287 dir path=var/spool/cron/crontabs group=sys
288 dir path=var/spool/locks group=uucp owner=uucp
289 dir path=var/svc group=sys
290 dir path=var/svc/log group=sys
291 dir path=var/svc/manifest group=sys
292 dir path=var/svc/manifest/application group=sys
293 dir path=var/svc/manifest/application/management group=sys
294 dir path=var/svc/manifest/application/print group=sys
295 dir path=var/svc/manifest/application/security group=sys
296 dir path=var/svc/manifest/device group=sys
297 dir path=var/svc/manifest/milestone group=sys
298 dir path=var/svc/manifest/network group=sys
299 dir path=var/svc/manifest/network/dns group=sys
300 dir path=var/svc/manifest/network/ipsec group=sys
301 dir path=var/svc/manifest/network/ldap group=sys
302 dir path=var/svc/manifest/network/nfs group=sys
303 dir path=var/svc/manifest/network/nis group=sys
304 dir path=var/svc/manifest/network/routing group=sys
305 dir path=var/svc/manifest/network/rpc group=sys
306 dir path=var/svc/manifest/network/security group=sys
307 dir path=var/svc/manifest/network/shares group=sys
308 dir path=var/svc/manifest/network/ssl group=sys
309 dir path=var/svc/manifest/platform group=sys
310 $(sparc_ONLY)dir path=var/svc/manifest/platform/sun4u group=sys
311 $(sparc_ONLY)dir path=var/svc/manifest/platform/sun4v group=sys
312 dir path=var/svc/manifest/site group=sys
313 dir path=var/svc/manifest/system group=sys
314 dir path=var/svc/manifest/system/device group=sys
315 dir path=var/svc/manifest/system/filesystem group=sys
316 dir path=var/svc/manifest/system/security group=sys
317 dir path=var/svc/manifest/system/svc group=sys
318 dir path=var/svc/profile group=sys
319 dir path=var/tmp group=sys mode=1777
320 driver name=dump perms="dump 0660 root sys"
321 driver name=fssnap \
322     policy="ctl read_priv_set=sys_config write_priv_set=sys_config" \
323     perms="* 0640 root sys" perms="ctl 0666 root sys"
324 driver name=kstat perms="* 0666 root sys"
325 driver name=ksyms perms="* 0666 root sys"

```

```

326 driver name=logindmux
327 driver name=ptm clone_perms="ptmx 0666 root sys"
328 driver name=pts perms="* 0644 root sys" perms="0 0620 root tty" \
329     perms="1 0620 root tty" perms="2 0620 root tty" perms="3 0620 root tty"
330 file path=etc/.login group=sys preserve=renamew
331 file path=etc/cron.d/.proto group=sys mode=0744
332 file path=etc/cron.d/at.deny group=sys preserve=true
333 file path=etc/cron.d/cron.deny group=sys preserve=true
334 file path=etc/cron.d/queuedefs group=sys
335 file path=etc/crypto/kmf.conf group=sys preserve=true
336 file path=etc/crypto/pkcs11.conf group=sys preserve=true
337 file path=etc/datensk group=sys mode=0444
338 file path=etc/default/cron group=sys preserve=true
339 file path=etc/default/devfsadm group=sys preserve=true
340 file path=etc/default/fs group=sys preserve=true
341 file path=etc/default/init group=sys preserve=true
342 file path=etc/default/keyerv group=sys preserve=true
343 file path=etc/default/login group=sys preserve=true
344 file path=etc/default/nss group=sys preserve=true
345 file path=etc/default/passwd group=sys preserve=true
346 file path=etc/default/su group=sys preserve=true
347 file path=etc/default/syslogd group=sys preserve=true
348 file path=etc/default/tar group=sys preserve=true
349 file path=etc/default/utmpd group=sys preserve=true
350 file path=etc/dev/reserved_devnames group=sys preserve=true
351 file path=etc/device.tab group=root mode=0444 preserve=true
352 file path=etc/dfs/dfstab group=sys preserve=true
353 file path=etc/dfs/fstypes group=root preserve=true
354 file path=etc/dfs/sharetab group=root mode=0444 preserve=true
355 file path=etc/dgroup.tab group=sys mode=0444 preserve=true
356 file path=etc/dhcp/inittab group=sys preserve=true
357 file path=etc/dhcp/inittab6 group=sys preserve=true
358 file path=etc/dumpdates group=sys mode=0664 preserve=true
359 file path=etc/format.dat group=sys preserve=true
360 file path=etc/fs/dev/mount mode=0555
361 file path=etc/fs/hsfs/mount mode=0555
362 file path=etc/fs/ufs/mount mode=0555
363 file path=etc/ftpd/ftpusers group=sys preserve=true
364 file path=etc/group group=sys preserve=true
365 file path=etc/inet/hosts group=sys preserve=true
366 file path=etc/inet/inetd.conf group=sys preserve=true
367 file path=etc/inet/ipaddrsl.conf group=sys preserve=true
368 file path=etc/inet/netmasks group=sys preserve=true
369 file path=etc/inet/networks group=sys preserve=true
370 file path=etc/inet/protocols group=sys preserve=true
371 file path=etc/inet/services group=sys preserve=true
372 file path=etc/inet/wanboot.conf.sample group=sys mode=0444
373 file path=etc/init.d/PRESERVE group=sys mode=0744 preserve=true
374 file path=etc/init.d/README group=sys preserve=true
375 file path=etc/init.d/cacheefs.daemon group=sys mode=0744 preserve=true
376 file path=etc/init.d/ldap.client group=sys mode=0744
377 file path=etc/init.d/nsd group=sys mode=0744
378 file path=etc/init.d/sysetup group=sys mode=0744 preserve=true
379 file path=etc/init.d/ufs_quota group=sys mode=0744 preserve=true
380 file path=etc/init.d/urcu group=sys preserve=true
381 file path=etc/ioctl.syscon group=sys preserve=true
382 file path=etc/ksh.kshrc group=sys preserve=renameld
383 file path=etc/logadm.conf group=sys preserve=true timestamp=19700101T000000Z
384 file path=etc/logindevperm group=sys preserve=true
385 file path=etc/magic mode=0444
386 file path=etc/mail/mailx.rc preserve=true
387 file path=etc/mailcap preserve=true
388 file path=etc/mime.types preserve=true
389 file path=etc/mnttab group=root mode=0444 preserve=true
390 file path=etc/motd group=sys preserve=true
391 file path=etc/net/ticlts/hosts group=sys

```

```
392 file path=etc/net/ticlts/services group=sys preserve=true
393 file path=etc/net/ticots/hosts group=sys
394 file path=etc/net/ticots/services group=sys preserve=true
395 file path=etc/net/ticotsord/hosts group=sys
396 file path=etc/net/ticotsord/services group=sys preserve=true
397 file path=etc/netconfig group=sys preserve=true
398 file path=etc/nscd.conf group=sys preserve=true
399 file path=etc/nsswitch.ad group=sys
400 file path=etc/nsswitch.conf group=sys preserve=true
401 file path=etc/nsswitch.dns group=sys
402 file path=etc/nsswitch.files group=sys
403 file path=etc/nsswitch.ldap group=sys
404 file path=etc/pam.conf group=sys preserve=true
405 file path=etc/passwd group=sys preserve=true
406 file path=etc/profile group=sys preserve=true
407 file path=etc/project group=sys preserve=true
408 file path=etc/rc2.d/README group=sys
409 file path=etc/rc3.d/README group=sys
410 file path=etc/rcS.d/README group=sys
411 file path=etc/remote preserve=true
412 file path=etc/rpc group=sys preserve=true
413 file path=etc/saf/_sactab group=sys preserve=true
414 file path=etc/saf/_sysconfig group=sys preserve=true
415 file path=etc/saf/zsmon/_pmtab group=sys preserve=true
416 file path=etc/security/audit_class group=sys preserve=renamew
417 file path=etc/security/audit_event group=sys preserve=renamew
418 file path=etc/security/audit_warn group=sys mode=0740 preserve=renamew
419 file path=etc/security/auth_attr group=sys preserve=true \
420 timestamp=19700101T000000Z
421 file path=etc/security/auth_attr.d/SUNWcs group=sys
422 file path=etc/security/crypt.conf group=sys preserve=renamew
423 file path=etc/security/dev/audio mode=0400
424 file path=etc/security/dev/fd0 mode=0400
425 file path=etc/security/dev/sr0 mode=0400
426 file path=etc/security/dev/st0 mode=0400
427 file path=etc/security/dev/st1 mode=0400
428 file path=etc/security/exec_attr group=sys preserve=true \
429 timestamp=19700101T000000Z
430 file path=etc/security/exec_attr.d/SUNWcs group=sys
431 file path=etc/security/kmfpolicy.xml
432 file path=etc/security/lib/audio_clean group=sys mode=0555
433 file path=etc/security/lib/fd_clean group=sys mode=0555
434 file path=etc/security/lib/sr_clean group=sys mode=0555
435 file path=etc/security/lib/st_clean group=sys mode=0555
436 file path=etc/security/policy.conf group=sys preserve=true
437 file path=etc/security/priv_names group=sys preserve=renamew
438 file path=etc/security/prof_attr group=sys preserve=true \
439 timestamp=19700101T000000Z
440 file path=etc/security/prof_attr.d/SUNWcs group=sys
441 file path=etc/shadow group=sys mode=0400 preserve=true
442 file path=etc/skel/.profile group=other preserve=true
443 file path=etc/skel/local.cshrc group=sys preserve=true
444 file path=etc/skel/local.login group=sys preserve=true
445 file path=etc/skel/local.profile group=sys preserve=true
446 file path=etc/svc/profile/generic.limited_net.xml group=sys mode=0444
447 file path=etc/svc/profile/generic_open.xml group=sys mode=0444
448 file path=etc/svc/profile/inetd_generic.xml group=sys mode=0444
449 file path=etc/svc/profile/inetd_upgrade.xml group=sys mode=0444
450 file path=etc/svc/profile/ns_dns.xml group=sys mode=0444
451 file path=etc/svc/profile/ns_files.xml group=sys mode=0444
452 file path=etc/svc/profile/ns_ldap.xml group=sys mode=0444
453 file path=etc/svc/profile/ns_nis.xml group=sys mode=0444
454 file path=etc/svc/profile/ns_none.xml group=sys mode=0444
455 $(sparc_ONLY)file path=etc/svc/profile/platform_SUNW,SPARC-Enterprise.xml \
456 group=sys mode=0444
457 $(sparc_ONLY)file path=etc/svc/profile/platform_SUNW,Sun-Fire-15000.xml \
```

```
458 group=sys mode=0444
459 $(sparc_ONLY)file path=etc/svc/profile/platform_SUNW,Sun-Fire-880.xml \
460 group=sys mode=0444
461 $(sparc_ONLY)file path=etc/svc/profile/platform_SUNW,Sun-Fire.xml group=sys \
462 mode=0444
463 $(sparc_ONLY)file \
464 path=etc/svc/profile/platform_SUNW,Ultra-Enterprise-10000.xml group=sys \
465 mode=0444
466 $(sparc_ONLY)file \
467 path=etc/svc/profile/platform_SUNW,UltraSPARC-III-Netractor.xml group=sys \
468 mode=0444
469 file path=etc/svc/profile/platform_none.xml group=sys mode=0444
470 $(sparc_ONLY)file path=etc/svc/profile/platform_sun4v.xml group=sys mode=0444
471 file path=etc/sysevent/config/README group=sys mode=0444
472 file path=etc/sysevent/config/SUNW,EC_dr,ESC_dr_req,sysevent.conf group=sys
473 file path=etc/syslog.conf group=sys preserve=true
474 file path=etc/ttydefs group=sys preserve=true
475 file path=etc/ttysrch group=sys preserve=true
476 file path=etc/user_attr group=sys preserve=true timestamp=19700101T000000Z
477 file path=etc/user_attr.d/SUNWcs group=sys
478 file path=etc/vfstab group=sys preserve=true
479 file path=lib/inet/in.mpathd mode=0555
480 file path=lib/inet/ipmgmt mode=0555
481 file path=lib/inet/netcfgd mode=0555
482 file path=lib/inet/nwamd mode=0555
483 file path=lib/svc/bin/lsvcrun group=sys mode=0555
484 file path=lib/svc/bin/mfstscan group=sys mode=0555
485 file path=lib/svc/bin/restore_repository group=sys mode=0555
486 file path=lib/svc/bin/sqlite group=sys mode=0555
487 file path=lib/svc/bin/svc.configd group=sys mode=0555
488 file path=lib/svc/bin/svc.ipfd group=sys mode=0555
489 file path=lib/svc/bin/svc.startd group=sys mode=0555
490 file path=lib/svc/manifest/milestone/multi-user-server.xml group=sys mode=0444
491 file path=lib/svc/manifest/milestone/multi-user.xml group=sys mode=0444
492 file path=lib/svc/manifest/milestone/name-services.xml group=sys mode=0444
493 file path=lib/svc/manifest/milestone/network.xml group=sys mode=0444
494 file path=lib/svc/manifest/milestone/single-user.xml group=sys mode=0444
495 file path=lib/svc/manifest/milestone/sysconfig.xml group=sys mode=0444
496 file path=lib/svc/manifest/network/dlmgmt.xml group=sys mode=0444
497 file path=lib/svc/manifest/network/dns/client.xml group=sys mode=0444
498 file path=lib/svc/manifest/network/dns/install.xml group=sys mode=0444
499 file path=lib/svc/manifest/network/forwarding.xml group=sys mode=0444
500 file path=lib/svc/manifest/network/inetd-upgrade.xml group=sys mode=0444
501 file path=lib/svc/manifest/network/inetd.xml group=sys mode=0444
502 file path=lib/svc/manifest/network/ipsec/ike.xml group=sys mode=0444
503 file path=lib/svc/manifest/network/ipsec/ipsecalgs.xml group=sys mode=0444
504 file path=lib/svc/manifest/network/ipsec/manual-key.xml group=sys mode=0444
505 file path=lib/svc/manifest/network/ipsec/policy.xml group=sys mode=0444
506 file path=lib/svc/manifest/network/ldap/client.xml group=sys mode=0444
507 file path=lib/svc/manifest/network/network-initial.xml group=sys mode=0444
508 file path=lib/svc/manifest/network/network-install.xml group=sys mode=0444
509 file path=lib/svc/manifest/network/network-ipmgmt.xml group=sys mode=0444
510 file path=lib/svc/manifest/network/network-ipqos.xml group=sys mode=0444
511 file path=lib/svc/manifest/network/network-iptun.xml group=sys mode=0444
512 file path=lib/svc/manifest/network/network-location.xml group=sys mode=0444
513 file path=lib/svc/manifest/network/network-loopback.xml group=sys mode=0444
514 file path=lib/svc/manifest/network/network-netcfg.xml group=sys mode=0444
515 file path=lib/svc/manifest/network/network-netmask.xml group=sys mode=0444
516 file path=lib/svc/manifest/network/network-physical.xml group=sys mode=0444
517 file path=lib/svc/manifest/network/network-routing-setup.xml group=sys \
518 mode=0444
519 file path=lib/svc/manifest/network/network-service.xml group=sys mode=0444
520 file path=lib/svc/manifest/network/routing/legacy-routing.xml group=sys \
521 mode=0444
522 file path=lib/svc/manifest/network/rpc/bind.xml group=sys mode=0444
523 file path=lib/svc/manifest/network/rpc/keyserv.xml group=sys mode=0444
```



```
524 file path=lib/svc/manifest/network/shares/group.xml group=sys mode=0444
525 file path=lib/svc/manifest/network/shares/reparsed.xml group=sys mode=0444
526 file path=lib/svc/manifest/network/socket-filter-kssl.xml group=sys mode=0444
527 file path=lib/svc/manifest/network/ssl/kssl-proxy.xml group=sys mode=0444
528 file path=lib/svc/manifest/system/auditd.xml group=sys mode=0444
529 file path=lib/svc/manifest/system/auditset.xml group=sys mode=0444
530 file path=lib/svc/manifest/system/boot-archive-update.xml group=sys mode=0444
531 file path=lib/svc/manifest/system/boot-archive.xml group=sys mode=0444
532 file path=lib/svc/manifest/system/boot-config.xml group=sys mode=0444
533 file path=lib/svc/manifest/system/consadm.xml group=sys mode=0444
534 file path=lib/svc/manifest/system/console-login.xml group=sys mode=0444
535 file path=lib/svc/manifest/system/coreadm.xml group=sys mode=0444
536 file path=lib/svc/manifest/system/cron.xml group=sys mode=0444
537 file path=lib/svc/manifest/system/cryptosvc.xml group=sys mode=0444
538 file path=lib/svc/manifest/system/device/allocate.xml group=sys mode=0444
539 file path=lib/svc/manifest/system/device/devices-audio.xml group=sys mode=0444
540 file path=lib/svc/manifest/system/device/devices-local.xml group=sys mode=0444
541 file path=lib/svc/manifest/system/device/mpxio-upgrade.xml group=sys mode=0444
542 file path=lib/svc/manifest/system/early-manifest-import.xml group=sys \
543 mode=0444
544 file path=lib/svc/manifest/system/extended-accounting.xml group=sys mode=0444
545 file path=lib/svc/manifest/system/filesystem/local-fs.xml group=sys mode=0444
546 file path=lib/svc/manifest/system/filesystem/minimal-fs.xml group=sys \
547 mode=0444
548 file path=lib/svc/manifest/system/filesystem/root-fs.xml group=sys mode=0444
549 file path=lib/svc/manifest/system/filesystem/usr-fs.xml group=sys mode=0444
550 $(i386_ONLY)file path=lib/svc/manifest/system/hostid.xml group=sys mode=0444
551 file path=lib/svc/manifest/system/hotplug.xml group=sys mode=0444
552 file path=lib/svc/manifest/system/identity.xml group=sys mode=0444
553 file path=lib/svc/manifest/system/idmap.xml group=sys mode=0444
554 file path=lib/svc/manifest/system/keymap.xml group=sys mode=0444
555 file path=lib/svc/manifest/system/logadm-upgrade.xml group=sys mode=0444
556 file path=lib/svc/manifest/system/manifest-import.xml group=sys mode=0444
557 file path=lib/svc/manifest/system/name-service-cache.xml group=sys mode=0444
558 file path=lib/svc/manifest/system/pfexecd.xml group=sys mode=0444
559 file path=lib/svc/manifest/system/rbac.xml group=sys mode=0444
560 file path=lib/svc/manifest/system/rmtmpfiles.xml group=sys mode=0444
561 file path=lib/svc/manifest/system/sac.xml group=sys mode=0444
562 file path=lib/svc/manifest/system/svc/global.xml group=sys mode=0444
563 file path=lib/svc/manifest/system/svc/restarter.xml group=sys mode=0444
564 file path=lib/svc/manifest/system/system-log.xml group=sys mode=0444
565 file path=lib/svc/manifest/system/utmp.xml group=sys mode=0444
566 file path=lib/svc/manifest/system/vtdaemon.xml group=sys mode=0444
567 file path=lib/svc/method/boot-archive mode=0555
568 file path=lib/svc/method/boot-archive-update mode=0555
569 file path=lib/svc/method/console-login mode=0555
570 file path=lib/svc/method/devices-audio mode=0555
571 file path=lib/svc/method/devices-local mode=0555
572 file path=lib/svc/method/dns-install mode=0555
573 file path=lib/svc/method/fs-local mode=0555
574 file path=lib/svc/method/fs-minimal mode=0555
575 file path=lib/svc/method/fs-root mode=0555
576 file path=lib/svc/method/fs-usr mode=0555
577 file path=lib/svc/method/identity-domain mode=0555
578 file path=lib/svc/method/identity-node mode=0555
579 file path=lib/svc/method/inetd-upgrade mode=0555
580 file path=lib/svc/method/keymap mode=0555
581 file path=lib/svc/method/ldap-client mode=0555
582 file path=lib/svc/method/logadm-upgrade mode=0555
583 file path=lib/svc/method/manifest-import mode=0555
584 file path=lib/svc/method/mpxio-upgrade mode=0555
585 file path=lib/svc/method/net-init mode=0555
586 file path=lib/svc/method/net-install mode=0555
587 file path=lib/svc/method/net-ipmgmt mode=0555
588 file path=lib/svc/method/net-ipqos mode=0555
589 file path=lib/svc/method/net-iptun mode=0555
```

```
590 file path=lib/svc/method/net-loc mode=0555
591 file path=lib/svc/method/net-loopback mode=0555
592 file path=lib/svc/method/net-netmask mode=0555
593 file path=lib/svc/method/net-nwam mode=0555
594 file path=lib/svc/method/net-physical mode=0555
595 file path=lib/svc/method/net-routing-setup mode=0555
596 file path=lib/svc/method/net-svc mode=0555
597 file path=lib/svc/method/rmtmpfiles mode=0555
598 file path=lib/svc/method/rpc-bind mode=0555
599 file path=lib/svc/method/svc-allocate mode=0555
600 file path=lib/svc/method/svc-auditd mode=0555
601 file path=lib/svc/method/svc-auditset mode=0555
602 file path=lib/svc/method/svc-boot-config mode=0555
603 file path=lib/svc/method/svc-consadm mode=0555
604 file path=lib/svc/method/svc-cron mode=0555
605 file path=lib/svc/method/svc-dlmgmt mode=0555
606 file path=lib/svc/method/svc-forwarding mode=0555
607 $(i386_ONLY)file path=lib/svc/method/svc-hostid mode=0555
608 file path=lib/svc/method/svc-hotplug mode=0555
609 file path=lib/svc/method/svc-legacy-routing mode=0555
610 file path=lib/svc/method/svc-nscd mode=0555
611 file path=lib/svc/method/svc-rbac mode=0555
612 file path=lib/svc/method/svc-sockfilter mode=0555
613 file path=lib/svc/method/svc-utmpd mode=0555
614 file path=lib/svc/method/system-log mode=0555
615 file path=lib/svc/method/vtdaemon mode=0555
616 file path=lib/svc/method/yp mode=0555
617 # global.db is not needed in non-global zones, and it's pretty large.
618 file path=lib/svc/seed/global.db group=sys mode=0444 \
619 variant.opensolaris.zone=global
620 # symmetrically, nonglobal.db is not needed in global zones.
621 file path=lib/svc/seed/nonglobal.db group=sys mode=0444 \
622 variant.opensolaris.zone=nonglobal
623 file path=lib/svc/share/README mode=0444
624 file path=lib/svc/share/fs_include.sh mode=0444
625 file path=lib/svc/share/ipf_include.sh mode=0444
626 file path=lib/svc/share/mfsthistory mode=0444
627 file path=lib/svc/share/net_include.sh mode=0444
628 file path=lib/svc/share/routing_include.sh mode=0444
629 file path=lib/svc/share/smf_include.sh mode=0444
630 file path=root/.bashrc group=root preserve=true
631 file path=root/.profile group=root preserve=true
632 file path=sbin/autopush mode=0555
633 $(i386_ONLY)file path=sbin/biosdev mode=0555
634 file path=sbin/bootadm mode=0555
635 file path=sbin/cryptoadm mode=0555
636 file path=sbin/devprop mode=0555
637 file path=sbin/dhccpagent mode=0555
638 file path=sbin/dhccpinfo mode=0555
639 file path=sbin/dlmgmt mode=0555
640 file path=sbin/fdisk mode=0555
641 file path=sbin/fiocompress mode=0555
642 file path=sbin/hostconfig mode=0555
643 file path=sbin/ifconfig mode=0555
644 file path=sbin/ifparse mode=0555
645 file path=sbin/init group=sys mode=0555
646 $(i386_ONLY)file path=sbin/installgrub group=sys mode=0555
647 file path=sbin/impstat mode=0555
648 file path=sbin/mount mode=0555
649 file path=sbin/mountall group=sys mode=0555
650 file path=sbin/netstrategy mode=0555
651 file path=sbin/rc0 group=sys mode=0744
652 file path=sbin/rc1 group=sys mode=0744
653 file path=sbin/rc2 group=sys mode=0744
654 file path=sbin/rc3 group=sys mode=0744
655 file path=sbin/rcS group=sys mode=0744
```

656 file path=sbin/route mode=0555
657 file path=sbin/routadm mode=0555
658 file path=sbin/soconfig mode=0555
659 file path=sbin/su.static group=sys mode=0555
660 file path=sbin/sulogin mode=0555
661 file path=sbin/swapadd group=sys mode=0744
662 file path=sbin/sync mode=0555
663 file path=sbin/tzreload mode=0555
664 file path=sbin/uadmin group=sys mode=0555
665 file path=sbin/umount mode=0555
666 file path=sbin/umountall group=sys mode=0555
667 file path=sbin/uname mode=0555
668 file path=sbin/wusbadm mode=0555
669 file path=sbin/zonename mode=0555
670 \$(i386_ONLY)file path=usr/bin/\$(ARCH32)/amt mode=0555
671 file path=usr/bin/\$(ARCH32)/decrypt mode=0555
672 file path=usr/bin/\$(ARCH32)/digest mode=0555
673 file path=usr/bin/\$(ARCH32)/ksh93 mode=0555
674 \$(i386_ONLY)file path=usr/bin/\$(ARCH32)/newtask group=sys mode=4555
675 \$(i386_ONLY)file path=usr/bin/\$(ARCH32)/nohup mode=0555
676 \$(i386_ONLY)file path=usr/bin/\$(ARCH32)/prctl mode=0555
677 \$(i386_ONLY)file path=usr/bin/\$(ARCH32)/prstat mode=0555
678 \$(i386_ONLY)file path=usr/bin/\$(ARCH32)/ps mode=0555
679 file path=usr/bin/\$(ARCH32)/savecore mode=0555
680 \$(i386_ONLY)file path=usr/bin/\$(ARCH32)/setuname mode=0555
681 \$(i386_ONLY)file path=usr/bin/\$(ARCH32)/uptime mode=4555
682 file path=usr/bin/\$(ARCH64)/amt mode=0555
683 file path=usr/bin/\$(ARCH64)/crle mode=0555
684 file path=usr/bin/\$(ARCH64)/decrypt mode=0555
685 file path=usr/bin/\$(ARCH64)/digest mode=0555
686 file path=usr/bin/\$(ARCH64)/ksh93 mode=0555
687 file path=usr/bin/\$(ARCH64)/ls mode=0555
688 file path=usr/bin/\$(ARCH64)/moe mode=0555
689 file path=usr/bin/\$(ARCH64)/newtask group=sys mode=4555
690 file path=usr/bin/\$(ARCH64)/nohup mode=0555
691 file path=usr/bin/\$(ARCH64)/prctl mode=0555
692 file path=usr/bin/\$(ARCH64)/prstat mode=0555
693 file path=usr/bin/\$(ARCH64)/ps mode=0555
694 file path=usr/bin/\$(ARCH64)/savecore mode=0555
695 file path=usr/bin/\$(ARCH64)/setuname mode=0555
696 file path=usr/bin/\$(ARCH64)/uptime mode=4555
697 \$(i386_ONLY)file path=usr/bin/addbadsec mode=0555
698 file path=usr/bin/alias mode=0555
699 file path=usr/bin/amt mode=0555
700 file path=usr/bin/arch mode=0555
701 file path=usr/bin/at group=sys mode=4755
702 file path=usr/bin/atq group=sys mode=4755
703 file path=usr/bin/atrm group=sys mode=4755
704 file path=usr/bin/auths mode=0555
705 file path=usr/bin/basename mode=0555
706 file path=usr/bin/busstat mode=0555
707 file path=usr/bin/captaininfo mode=0555
708 file path=usr/bin/cat mode=0555
709 file path=usr/bin/chgrp mode=0555
710 file path=usr/bin/chmod mode=0555
711 file path=usr/bin/chown mode=0555
712 file path=usr/bin/ckdate mode=0555
713 file path=usr/bin/ckgid mode=0555
714 file path=usr/bin/ckint mode=0555
715 file path=usr/bin/ckitem mode=0555
716 file path=usr/bin/ckkeywd mode=0555
717 file path=usr/bin/ckpath mode=0555
718 file path=usr/bin/ckrange mode=0555
719 file path=usr/bin/ckstr mode=0555
720 file path=usr/bin/cktime mode=0555
721 file path=usr/bin/ckuid mode=0555

722 file path=usr/bin/ckyorn mode=0555
723 file path=usr/bin/clear mode=0555
724 file path=usr/bin/coreadm mode=0555
725 file path=usr/bin/cp mode=0555
726 file path=usr/bin/cpio mode=0555
727 file path=usr/bin/crle mode=0555
728 file path=usr/bin/crontab mode=4555
729 file path=usr/bin/crypt mode=0555
730 file path=usr/bin/csh mode=0555
731 file path=usr/bin/ctrunc mode=0555
732 file path=usr/bin/ctstat mode=0555
733 file path=usr/bin/ctwatch mode=0555
734 file path=usr/bin/date mode=0555
735 file path=usr/bin/dd mode=0555
736 file path=usr/bin/devattr mode=0555
737 file path=usr/bin/devfree mode=0555
738 file path=usr/bin/devreserv mode=0555
739 file path=usr/bin/dirname mode=0555
740 \$(i386_ONLY)file path=usr/bin/diskscan mode=0555
741 file path=usr/bin/domainname mode=0555
742 file path=usr/bin/du mode=0555
743 file path=usr/bin/dumpcs mode=0555
744 file path=usr/bin/dumpkeys mode=0555
745 file path=usr/bin/echo mode=0555
746 file path=usr/bin/ed mode=0555
747 file path=usr/bin/egrep mode=0555
748 file path=usr/bin/eject mode=0555
749 file path=usr/bin/env mode=0555
750 file path=usr/bin/expr mode=0555
751 file path=usr/bin/false mode=0555
752 file path=usr/bin/fdetach mode=0555
753 file path=usr/bin/fdformat mode=4555
754 file path=usr/bin/fgrep mode=0555
755 file path=usr/bin/file mode=0555
756 file path=usr/bin/find mode=0555
757 file path=usr/bin/fmt mode=0555
758 file path=usr/bin/fmtmsg mode=0555
759 file path=usr/bin/fold mode=0555
760 file path=usr/bin/fsstat mode=0555
761 file path=usr/bin/geniconvtbl mode=0555
762 file path=usr/bin/getconf mode=0555
763 file path=usr/bin/getdev mode=0555
764 file path=usr/bin/getdgrp mode=0555
765 file path=usr/bin/getent mode=0555
766 file path=usr/bin/getfacl mode=0555
767 file path=usr/bin/getopt mode=0555
768 file path=usr/bin/gettext mode=0555
769 file path=usr/bin/getvol mode=0555
770 file path=usr/bin/grep mode=0555
771 file path=usr/bin/groups mode=0555
772 file path=usr/bin/head mode=0555
773 file path=usr/bin/hostid mode=0555
774 file path=usr/bin/hostname mode=0555
775 file path=usr/bin/i286 mode=0555
776 file path=usr/bin/iconv mode=0555
777 file path=usr/bin/id mode=0555
778 file path=usr/bin/infocmp mode=0555
779 file path=usr/bin/iostat mode=0555
780 file path=usr/bin/isainfo mode=0555
781 file path=usr/bin/isalist mode=0555
782 file path=usr/bin/kbd mode=0555
783 file path=usr/bin/keylogin mode=0555
784 file path=usr/bin/keylogout mode=0555
785 file path=usr/bin/kmfcfg mode=0555
786 file path=usr/bin/kvmstat mode=0555
787 file path=usr/bin/line mode=0555

```

788 file path=usr/bin/listdgrp mode=0555
789 file path=usr/bin/listusers mode=0555
790 file path=usr/bin/loadkeys mode=0555
791 file path=usr/bin/logger mode=0555
792 file path=usr/bin/login mode=4555
793 file path=usr/bin/logins mode=0750
794 file path=usr/bin/ls mode=0555
795 file path=usr/bin/m4 mode=0555
796 file path=usr/bin/mach mode=0555
797 file path=usr/bin/mail group=mail mode=2511
798 file path=usr/bin/mailx group=mail mode=2511
799 file path=usr/bin/makedev mode=0555
800 file path=usr/bin/mesg mode=0555
801 file path=usr/bin/mkdir mode=0555
802 file path=usr/bin/mkpwdict mode=0555
803 file path=usr/bin/mktemp mode=0555
804 file path=usr/bin/moe mode=0555
805 file path=usr/bin/more mode=0555
806 file path=usr/bin/mpstat mode=0555
807 file path=usr/bin/mt mode=0555
808 file path=usr/bin/netstat mode=0555
809 file path=usr/bin/newgrp group=sys mode=4755
810 file path=usr/bin/nice mode=0555
811 file path=usr/bin/optisa mode=0555
812 file path=usr/bin/pagesize mode=0555
813 file path=usr/bin/passwd group=sys mode=6555
814 file path=usr/bin/pathchk mode=0555
815 file path=usr/bin/pax mode=0555
816 file path=usr/bin/pfexec mode=0555
817 file path=usr/bin/pg mode=0555
818 file path=usr/bin/pgrep mode=0555
819 file path=usr/bin/pktool mode=0555
820 file path=usr/bin/pr mode=0555
821 file path=usr/bin/printf mode=0555
822 file path=usr/bin/priocntl mode=0555
823 file path=usr/bin/profiles mode=0555
824 file path=usr/bin/projects mode=0555
825 file path=usr/bin/putdev mode=0555
826 file path=usr/bin/putdgrp mode=0555
827 file path=usr/bin/pwd mode=0555
828 file path=usr/bin/renice mode=0555
829 file path=usr/bin/rm mode=0555
830 file path=usr/bin/rmdir mode=0555
831 file path=usr/bin/roles mode=0555
832 file path=usr/bin/rpcinfo mode=0555
833 file path=usr/bin/runat mode=0555
834 file path=usr/bin/script mode=0555
835 file path=usr/bin/sed mode=0555
836 file path=usr/bin/setfacl mode=0555
837 file path=usr/bin/setpggrp group=sys mode=0555
838 file path=usr/bin/settime mode=0555
839 file path=usr/bin/shcomp mode=0555
840 file path=usr/bin/strchg group=root mode=0555
841 file path=usr/bin/strconf group=root mode=0555
842 file path=usr/bin/stty mode=0555
843 file path=usr/bin/su group=sys mode=4555
844 file path=usr/bin/svcprop mode=0555
845 file path=usr/bin/svcs mode=0555
846 file path=usr/bin/tabs mode=0555
847 file path=usr/bin/tail mode=0555
848 file path=usr/bin/tic mode=0555
849 file path=usr/bin/time mode=0555
850 file path=usr/bin/tip mode=4511 owner=uucp
851 file path=usr/bin/tpmadm mode=0555
852 file path=usr/bin/tput mode=0555
853 file path=usr/bin/tr mode=0555

```

```

854 file path=usr/bin/true mode=0555
855 file path=usr/bin/tty mode=0555
856 file path=usr/bin/tzselect mode=0555
857 file path=usr/bin/userattr mode=0555
858 file path=usr/bin/vmstat mode=0555
859 file path=usr/bin/which mode=0555
860 file path=usr/bin/who mode=0555
861 file path=usr/bin/wracct mode=0555
862 file path=usr/bin/write group=tty mode=2555
863 file path=usr/bin/xargs mode=0555
864 file path=usr/bin/xstr mode=0555
865 file path=usr/has/bin/edit mode=0555
866 file path=usr/has/bin/sh mode=0555
867 file path=usr/has/man/man1has/edit.1has
868 file path=usr/has/man/man1has/ex.1has
869 file path=usr/has/man/man1has/sh.1has
870 file path=usr/has/man/man1has/vi.1has
871 file path=usr/kernel/drv/$(ARCH64)/dump group=sys
872 file path=usr/kernel/drv/$(ARCH64)/fssnap group=sys
873 file path=usr/kernel/drv/$(ARCH64)/kstat group=sys
874 file path=usr/kernel/drv/$(ARCH64)/ksyms group=sys
875 file path=usr/kernel/drv/$(ARCH64)/logindmux group=sys
876 file path=usr/kernel/drv/$(ARCH64)/ptm group=sys
877 file path=usr/kernel/drv/$(ARCH64)/pts group=sys
878 $(i386_ONLY)file path=usr/kernel/drv/dump group=sys
879 file path=usr/kernel/drv/dump.conf group=sys
880 $(i386_ONLY)file path=usr/kernel/drv/fssnap group=sys
881 file path=usr/kernel/drv/fssnap.conf group=sys
882 $(i386_ONLY)file path=usr/kernel/drv/kstat group=sys
883 file path=usr/kernel/drv/kstat.conf group=sys
884 $(i386_ONLY)file path=usr/kernel/drv/ksyms group=sys
885 file path=usr/kernel/drv/ksyms.conf group=sys
886 $(i386_ONLY)file path=usr/kernel/drv/logindmux group=sys
887 file path=usr/kernel/drv/logindmux.conf group=sys
888 $(i386_ONLY)file path=usr/kernel/drv/ptm group=sys
889 file path=usr/kernel/drv/ptm.conf group=sys
890 $(i386_ONLY)file path=usr/kernel/drv/pts group=sys
891 file path=usr/kernel/drv/pts.conf group=sys
892 file path=usr/kernel/exec/$(ARCH64)/javaexec group=sys mode=0755
893 file path=usr/kernel/exec/$(ARCH64)/shbinexec group=sys mode=0755
894 $(i386_ONLY)file path=usr/kernel/exec/javaexec group=sys mode=0755
895 $(i386_ONLY)file path=usr/kernel/exec/shbinexec group=sys mode=0755
896 file path=usr/kernel/fs/$(ARCH64)/fdfs group=sys mode=0755
897 file path=usr/kernel/fs/$(ARCH64)/pcfs group=sys mode=0755
898 $(i386_ONLY)file path=usr/kernel/fs/fdfs group=sys mode=0755
899 $(i386_ONLY)file path=usr/kernel/fs/pcfs group=sys mode=0755
900 file path=usr/kernel/sched/$(ARCH64)/FX group=sys mode=0755
901 file path=usr/kernel/sched/$(ARCH64)/FX_DPTBL group=sys mode=0755
902 file path=usr/kernel/sched/$(ARCH64)/IA group=sys mode=0755
903 file path=usr/kernel/sched/$(ARCH64)/RT group=sys mode=0755
904 file path=usr/kernel/sched/$(ARCH64)/RT_DPTBL group=sys mode=0755
905 $(i386_ONLY)file path=usr/kernel/sched/FX group=sys mode=0755
906 $(i386_ONLY)file path=usr/kernel/sched/FX_DPTBL group=sys mode=0755
907 $(i386_ONLY)file path=usr/kernel/sched/IA group=sys mode=0755
908 $(i386_ONLY)file path=usr/kernel/sched/RT group=sys mode=0755
909 $(i386_ONLY)file path=usr/kernel/sched/RT_DPTBL group=sys mode=0755
910 file path=usr/kernel/strmod/$(ARCH64)/cryptmod group=sys mode=0755
911 file path=usr/kernel/strmod/$(ARCH64)/rlmod group=sys mode=0755
912 file path=usr/kernel/strmod/$(ARCH64)/telmod group=sys mode=0755
913 $(i386_ONLY)file path=usr/kernel/strmod/cryptmod group=sys mode=0755
914 $(i386_ONLY)file path=usr/kernel/strmod/rlmod group=sys mode=0755
915 $(i386_ONLY)file path=usr/kernel/strmod/telmod group=sys mode=0755
916 file path=usr/kernel/sys/$(ARCH64)/acctctl group=sys mode=0755
917 file path=usr/kernel/sys/$(ARCH64)/exacctsys group=sys mode=0755
918 file path=usr/kernel/sys/$(ARCH64)/sysacct group=sys mode=0755
919 $(i386_ONLY)file path=usr/kernel/sys/acctctl group=sys mode=0755

```

```

920 $(i386_ONLY)file path=usr/kernel/sys/exacctsys group=sys mode=0755
921 $(i386_ONLY)file path=usr/kernel/sys/sysacct group=sys mode=0755
922 file path=usr/kvm/README group=sys
923 file path=usr/lib/$(ARCH64)/libshare.so.1
924 file path=usr/lib/audit/audit_record_attr mode=0444
925 file path=usr/lib/calprog mode=0555
926 file path=usr/lib/class/FX/FXdispadmin mode=0555
927 file path=usr/lib/class/FX/FXpriocntl mode=0555
928 file path=usr/lib/class/IA/IAdispadmin mode=0555
929 file path=usr/lib/class/IA/IApriocntl mode=0555
930 file path=usr/lib/class/RT/RTdispadmin mode=0555
931 file path=usr/lib/class/RT/RTpriocntl mode=0555
932 file path=usr/lib/class/SDC/SDCdispadmin mode=0555
933 file path=usr/lib/class/SDC/SDCpriocntl mode=0555
934 file path=usr/lib/class/TS/TSdispadmin mode=0555
935 file path=usr/lib/class/TS/TSpriocntl mode=0555
936 file path=usr/lib/devfsadm/linkmod/SUNW_audio_link.so group=sys
937 file path=usr/lib/devfsadm/linkmod/SUNW_cfg_link.so group=sys
938 file path=usr/lib/devfsadm/linkmod/SUNW_disk_link.so group=sys
939 file path=usr/lib/devfsadm/linkmod/SUNW_fssnap_link.so group=sys
940 file path=usr/lib/devfsadm/linkmod/SUNW_ieee1394_link.so group=sys
941 file path=usr/lib/devfsadm/linkmod/SUNW_lofi_link.so group=sys
942 file path=usr/lib/devfsadm/linkmod/SUNW_md_link.so group=sys
943 file path=usr/lib/devfsadm/linkmod/SUNW_misc_link.so group=sys
944 file path=usr/lib/devfsadm/linkmod/SUNW_misc_link_$(ARCH).so group=sys
945 file path=usr/lib/devfsadm/linkmod/SUNW_port_link.so group=sys
946 file path=usr/lib/devfsadm/linkmod/SUNW_ramdisk_link.so group=sys
947 file path=usr/lib/devfsadm/linkmod/SUNW_sgen_link.so group=sys
948 file path=usr/lib/devfsadm/linkmod/SUNW_smp_link.so group=sys
949 file path=usr/lib/devfsadm/linkmod/SUNW_tape_link.so group=sys
950 file path=usr/lib/devfsadm/linkmod/SUNW_usb_link.so group=sys
951 $(i386_ONLY)file path=usr/lib/devfsadm/linkmod/SUNW_xen_link.so group=sys
952 file path=usr/lib/diffh mode=0555
953 file path=usr/lib/expreserve mode=0555
954 file path=usr/lib/exrecover mode=0555
955 file path=usr/lib/fs/cachefs/cachefs mode=0555
956 file path=usr/lib/fs/cachefs/cachefslog mode=0555
957 file path=usr/lib/fs/cachefs/cachefspack mode=0555
958 file path=usr/lib/fs/cachefs/cachefsstat mode=0555
959 file path=usr/lib/fs/cachefs/cachefswssize mode=0555
960 file path=usr/lib/fs/cachefs/cfsadmin mode=0555
961 file path=usr/lib/fs/cachefs/cfsfstype mode=0555
962 file path=usr/lib/fs/cachefs/cfstagchk mode=0555
963 file path=usr/lib/fs/cachefs/dfshares mode=0555
964 file path=usr/lib/fs/cachefs/fsck mode=0555
965 file path=usr/lib/fs/cachefs/mount mode=0555
966 file path=usr/lib/fs/cachefs/share mode=0555
967 file path=usr/lib/fs/cachefs/umount mode=0555
968 file path=usr/lib/fs/cachefs/unshare mode=0555
969 file path=usr/lib/fs/ctfs/mount mode=0555
970 file path=usr/lib/fs/fd/mount mode=0555
971 file path=usr/lib/fs/hsfs/fstyp.so.1 mode=0555
972 file path=usr/lib/fs/hsfs/labelit mode=0555
973 file path=usr/lib/fs/lofs/mount mode=0555
974 file path=usr/lib/fs/mntfs/mount mode=0555
975 file path=usr/lib/fs/objfs/mount mode=0555
976 file path=usr/lib/fs/proc/mount mode=0555
977 file path=usr/lib/fs/shares/mount mode=0555
978 file path=usr/lib/fs/tmpfs/mount mode=0555
979 file path=usr/lib/fs/ufs/clri mode=0555
980 file path=usr/lib/fs/ufs/df mode=0555
981 file path=usr/lib/fs/ufs/edquota mode=0555
982 file path=usr/lib/fs/ufs/ff mode=0555
983 file path=usr/lib/fs/ufs/fsck mode=0555
984 file path=usr/lib/fs/ufs/fsckall mode=0555
985 file path=usr/lib/fs/ufs/fsdb mode=0555

```

```

986 file path=usr/lib/fs/ufs/fsirand mode=0555
987 file path=usr/lib/fs/ufs/fssnap mode=0555
988 file path=usr/lib/fs/ufs/fstyp.so.1 mode=0555
989 file path=usr/lib/fs/ufs/labelit mode=0555
990 file path=usr/lib/fs/ufs/lockfs mode=0555
991 file path=usr/lib/fs/ufs/mkfs mode=0555
992 file path=usr/lib/fs/ufs/ncheck mode=0555
993 file path=usr/lib/fs/ufs/newfs mode=0555
994 file path=usr/lib/fs/ufs/quot mode=0555
995 file path=usr/lib/fs/ufs/quota mode=4555
996 file path=usr/lib/fs/ufs/quotacheck mode=0555
997 file path=usr/lib/fs/ufs/quotaooff mode=0555
998 file path=usr/lib/fs/ufs/repquota mode=0555
999 file path=usr/lib/fs/ufs/tunefs mode=0555
1000 file path=usr/lib/fs/ufs/ufsdump mode=4555
1001 file path=usr/lib/fs/ufs/ufsrestore mode=4555
1002 file path=usr/lib/fs/ufs/volcopy mode=0555
1003 file path=usr/lib/getoptcvt mode=0555
1004 file path=usr/lib/help/auths/locale/C/AllSolAuthsHeader.html
1005 file path=usr/lib/help/auths/locale/C/AuditHeader.html
1006 file path=usr/lib/help/auths/locale/C/AuthJobsAdmin.html
1007 file path=usr/lib/help/auths/locale/C/AuthJobsUser.html
1008 file path=usr/lib/help/auths/locale/C/AuthProfmgrAssign.html
1009 file path=usr/lib/help/auths/locale/C/AuthProfmgrDelegate.html
1010 file path=usr/lib/help/auths/locale/C/AuthProfmgrExecattrWrite.html
1011 file path=usr/lib/help/auths/locale/C/AuthProfmgrRead.html
1012 file path=usr/lib/help/auths/locale/C/AuthProfmgrWrite.html
1013 file path=usr/lib/help/auths/locale/C/AuthReadNDMP.html
1014 file path=usr/lib/help/auths/locale/C/AuthReadSMB.html
1015 file path=usr/lib/help/auths/locale/C/AuthRoleAssign.html
1016 file path=usr/lib/help/auths/locale/C/AuthRoleDelegate.html
1017 file path=usr/lib/help/auths/locale/C/AuthRoleWrite.html
1018 file path=usr/lib/help/auths/locale/C/BindStates.html
1019 file path=usr/lib/help/auths/locale/C/DevAllocHeader.html
1020 file path=usr/lib/help/auths/locale/C/DevAllocate.html
1021 file path=usr/lib/help/auths/locale/C/DevConfig.html
1022 file path=usr/lib/help/auths/locale/C/DevGrant.html
1023 file path=usr/lib/help/auths/locale/C/DevRevoke.html
1024 file path=usr/lib/help/auths/locale/C/DhccpmgrHeader.html
1025 file path=usr/lib/help/auths/locale/C/DhccpmgrWrite.html
1026 file path=usr/lib/help/auths/locale/C/HotplugHeader.html
1027 file path=usr/lib/help/auths/locale/C/HotplugModify.html
1028 file path=usr/lib/help/auths/locale/C/IdmapRules.html
1029 file path=usr/lib/help/auths/locale/C/JobHeader.html
1030 file path=usr/lib/help/auths/locale/C/JobsGrant.html
1031 file path=usr/lib/help/auths/locale/C/LinkSecurity.html
1032 file path=usr/lib/help/auths/locale/C/LoginEnable.html
1033 file path=usr/lib/help/auths/locale/C/LoginHeader.html
1034 file path=usr/lib/help/auths/locale/C/LoginRemote.html
1035 file path=usr/lib/help/auths/locale/C/NetworkAutoconfRead.html
1036 file path=usr/lib/help/auths/locale/C/NetworkAutoconfSelect.html
1037 file path=usr/lib/help/auths/locale/C/NetworkAutoconfWlan.html
1038 file path=usr/lib/help/auths/locale/C/NetworkAutoconfWrite.html
1039 file path=usr/lib/help/auths/locale/C/NetworkHeader.html
1040 file path=usr/lib/help/auths/locale/C/NetworkILBconf.html
1041 file path=usr/lib/help/auths/locale/C/NetworkILBenable.html
1042 file path=usr/lib/help/auths/locale/C/NetworkInterfaceConfig.html
1043 file path=usr/lib/help/auths/locale/C/NetworkVRRP.html
1044 file path=usr/lib/help/auths/locale/C/PriAdmin.html
1045 file path=usr/lib/help/auths/locale/C/ProfmgrHeader.html
1046 file path=usr/lib/help/auths/locale/C/RoleHeader.html
1047 file path=usr/lib/help/auths/locale/C/SmfAllLocate.html
1048 file path=usr/lib/help/auths/locale/C/SmfAutofsStates.html
1049 file path=usr/lib/help/auths/locale/C/SmfCoreadmStates.html
1050 file path=usr/lib/help/auths/locale/C/SmfCronStates.html
1051 file path=usr/lib/help/auths/locale/C/SmfExecctFlowStates.html

```

1052 file path=usr/lib/help/auths/locale/C/SmfExAcctNetStates.html
1053 file path=usr/lib/help/auths/locale/C/SmfExAcctProcessStates.html
1054 file path=usr/lib/help/auths/locale/C/SmfExAcctTaskStates.html
1055 file path=usr/lib/help/auths/locale/C/SmfHeader.html
1056 file path=usr/lib/help/auths/locale/C/SmfILBStates.html
1057 file path=usr/lib/help/auths/locale/C/SmfIPsecStates.html
1058 file path=usr/lib/help/auths/locale/C/SmfIdmapStates.html
1059 file path=usr/lib/help/auths/locale/C/SmfInetdStates.html
1060 file path=usr/lib/help/auths/locale/C/SmfLocationStates.html
1061 file path=usr/lib/help/auths/locale/C/SmfMDNSStates.html
1062 file path=usr/lib/help/auths/locale/C/SmfManageAudit.html
1063 file path=usr/lib/help/auths/locale/C/SmfManageHeader.html
1064 file path=usr/lib/help/auths/locale/C/SmfManageHotplug.html
1065 file path=usr/lib/help/auths/locale/C/SmfManageZFSnap.html
1066 file path=usr/lib/help/auths/locale/C/SmfModifyAppl.html
1067 file path=usr/lib/help/auths/locale/C/SmfModifyDepend.html
1068 file path=usr/lib/help/auths/locale/C/SmfModifyFramework.html
1069 file path=usr/lib/help/auths/locale/C/SmfModifyHeader.html
1070 file path=usr/lib/help/auths/locale/C/SmfModifyMethod.html
1071 file path=usr/lib/help/auths/locale/C/SmfNADDStates.html
1072 file path=usr/lib/help/auths/locale/C/SmfNDMPStates.html
1073 file path=usr/lib/help/auths/locale/C/SmfNWAMStates.html
1074 file path=usr/lib/help/auths/locale/C/SmfNscdStates.html
1075 file path=usr/lib/help/auths/locale/C/SmfPowerStates.html
1076 file path=usr/lib/help/auths/locale/C/SmfReparseStates.html
1077 file path=usr/lib/help/auths/locale/C/SmfRoutingStates.html
1078 file path=usr/lib/help/auths/locale/C/SmfSMBFSStates.html
1079 file path=usr/lib/help/auths/locale/C/SmfSMBStates.html
1080 file path=usr/lib/help/auths/locale/C/SmfSendmailStates.html
1081 file path=usr/lib/help/auths/locale/C/SmfSshStates.html
1082 file path=usr/lib/help/auths/locale/C/SmfSyslogStates.html
1083 file path=usr/lib/help/auths/locale/C/SmfVRRPStates.html
1084 file path=usr/lib/help/auths/locale/C/SmfValueAudit.html
1085 file path=usr/lib/help/auths/locale/C/SmfValueCoreadm.html
1086 file path=usr/lib/help/auths/locale/C/SmfValueExAcctFlow.html
1087 file path=usr/lib/help/auths/locale/C/SmfValueExAcctNet.html
1088 file path=usr/lib/help/auths/locale/C/SmfValueExAcctProcess.html
1089 file path=usr/lib/help/auths/locale/C/SmfValueExAcctTask.html
1090 file path=usr/lib/help/auths/locale/C/SmfValueFirewall.html
1091 file path=usr/lib/help/auths/locale/C/SmfValueHeader.html
1092 file path=usr/lib/help/auths/locale/C/SmfValueIPsec.html
1093 file path=usr/lib/help/auths/locale/C/SmfValueIdmap.html
1094 file path=usr/lib/help/auths/locale/C/SmfValueInetd.html
1095 file path=usr/lib/help/auths/locale/C/SmfValueMDNS.html
1096 file path=usr/lib/help/auths/locale/C/SmfValueNADD.html
1097 file path=usr/lib/help/auths/locale/C/SmfValueNDMP.html
1098 file path=usr/lib/help/auths/locale/C/SmfValueNWAM.html
1099 file path=usr/lib/help/auths/locale/C/SmfValueRouting.html
1100 file path=usr/lib/help/auths/locale/C/SmfValueSMB.html
1101 file path=usr/lib/help/auths/locale/C/SmfValueVscan.html
1102 file path=usr/lib/help/auths/locale/C/SmfValueVt.html
1103 file path=usr/lib/help/auths/locale/C/SmfVscanStates.html
1104 file path=usr/lib/help/auths/locale/C/SmfVtStates.html
1105 file path=usr/lib/help/auths/locale/C/SmfWpaStates.html
1106 file path=usr/lib/help/auths/locale/C/SysCpuPowerMgmt.html
1107 file path=usr/lib/help/auths/locale/C/SysDate.html
1108 file path=usr/lib/help/auths/locale/C/SysHeader.html
1109 file path=usr/lib/help/auths/locale/C/SysMaintenance.html
1110 file path=usr/lib/help/auths/locale/C/SysPowerMgmtBrightness.html
1111 file path=usr/lib/help/auths/locale/C/SysPowerMgmtHeader.html
1112 file path=usr/lib/help/auths/locale/C/SysPowerMgmtSuspend.html
1113 file path=usr/lib/help/auths/locale/C/SysPowerMgmtSuspendtoDisk.html
1114 file path=usr/lib/help/auths/locale/C/SysPowerMgmtSuspendtoRAM.html
1115 file path=usr/lib/help/auths/locale/C/SysShutdown.html
1116 file path=usr/lib/help/auths/locale/C/SysSyseventRead.html
1117 file path=usr/lib/help/auths/locale/C/SysSyseventWrite.html

1118 file path=usr/lib/help/auths/locale/C/WifiConfig.html
1119 file path=usr/lib/help/auths/locale/C/WifiWep.html
1120 file path=usr/lib/help/auths/locale/C/ZoneCloneFrom.html
1121 file path=usr/lib/help/auths/locale/C/ZoneHeader.html
1122 file path=usr/lib/help/auths/locale/C/ZoneLogin.html
1123 file path=usr/lib/help/auths/locale/C/ZoneManage.html
1124 file path=usr/lib/help/profiles/locale/C/RtAcctadm.html
1125 file path=usr/lib/help/profiles/locale/C/RtAll.html
1126 file path=usr/lib/help/profiles/locale/C/RtAuditCfg.html
1127 file path=usr/lib/help/profiles/locale/C/RtAuditCtrl.html
1128 file path=usr/lib/help/profiles/locale/C/RtAuditReview.html
1129 file path=usr/lib/help/profiles/locale/C/RtCPUPowerManagement.html
1130 file path=usr/lib/help/profiles/locale/C/RtConsUser.html
1131 file path=usr/lib/help/profiles/locale/C/RtContractObserver.html
1132 file path=usr/lib/help/profiles/locale/C/RtCronMngmnt.html
1133 file path=usr/lib/help/profiles/locale/C/RtCryptoMngmnt.html
1134 file path=usr/lib/help/profiles/locale/C/RtDHCPMngmnt.html
1135 file path=usr/lib/help/profiles/locale/C/RtDataAdmin.html
1136 file path=usr/lib/help/profiles/locale/C/RtDefault.html
1137 file path=usr/lib/help/profiles/locale/C/RtDeviceMngmnt.html
1138 file path=usr/lib/help/profiles/locale/C/RtDeviceSecurity.html
1139 file path=usr/lib/help/profiles/locale/C/RtExAcctFlow.html
1140 file path=usr/lib/help/profiles/locale/C/RtExAcctNet.html
1141 file path=usr/lib/help/profiles/locale/C/RtExAcctProcess.html
1142 file path=usr/lib/help/profiles/locale/C/RtExAcctTask.html
1143 file path=usr/lib/help/profiles/locale/C/RtFTPMngmnt.html
1144 file path=usr/lib/help/profiles/locale/C/RtFileSysMngmnt.html
1145 file path=usr/lib/help/profiles/locale/C/RtFileSysSecurity.html
1146 file path=usr/lib/help/profiles/locale/C/RtHotplugMngmnt.html
1147 file path=usr/lib/help/profiles/locale/C/RtIPFilterMngmnt.html
1148 file path=usr/lib/help/profiles/locale/C/RtIdmapMngmnt.html
1149 file path=usr/lib/help/profiles/locale/C/RtIdmapNameRulesMngmnt.html
1150 file path=usr/lib/help/profiles/locale/C/RtInetdMngmnt.html
1151 file path=usr/lib/help/profiles/locale/C/RtKerberosClntMngmnt.html
1152 file path=usr/lib/help/profiles/locale/C/RtKerberosSrvrMngmnt.html
1153 file path=usr/lib/help/profiles/locale/C/RtLogMngmnt.html
1154 file path=usr/lib/help/profiles/locale/C/RtMailMngmnt.html
1155 file path=usr/lib/help/profiles/locale/C/RtMaintAndRepair.html
1156 file path=usr/lib/help/profiles/locale/C/RtMediaBkup.html
1157 file path=usr/lib/help/profiles/locale/C/RtMediaCtlg.html
1158 file path=usr/lib/help/profiles/locale/C/RtMediaRestore.html
1159 file path=usr/lib/help/profiles/locale/C/RtNDMPMngmnt.html
1160 file path=usr/lib/help/profiles/locale/C/RtNameServiceAdmin.html
1161 file path=usr/lib/help/profiles/locale/C/RtNameServiceSecure.html
1162 file path=usr/lib/help/profiles/locale/C/RtNetAutoconfAdmin.html
1163 file path=usr/lib/help/profiles/locale/C/RtNetAutoconfUser.html
1164 file path=usr/lib/help/profiles/locale/C/RtNetILB.html
1165 file path=usr/lib/help/profiles/locale/C/RtNetIPsec.html
1166 file path=usr/lib/help/profiles/locale/C/RtNetLinkSecure.html
1167 file path=usr/lib/help/profiles/locale/C/RtNetMngmnt.html
1168 file path=usr/lib/help/profiles/locale/C/RtNetObservability.html
1169 file path=usr/lib/help/profiles/locale/C/RtNetSecure.html
1170 file path=usr/lib/help/profiles/locale/C/RtNetVRRP.html
1171 file path=usr/lib/help/profiles/locale/C/RtNetWifiMngmnt.html
1172 file path=usr/lib/help/profiles/locale/C/RtNetWifiSecure.html
1173 file path=usr/lib/help/profiles/locale/C/RtObAccessMngmnt.html
1174 file path=usr/lib/help/profiles/locale/C/RtOperator.html
1175 file path=usr/lib/help/profiles/locale/C/RtPriAdmin.html
1176 file path=usr/lib/help/profiles/locale/C/RtPrntAdmin.html
1177 file path=usr/lib/help/profiles/locale/C/RtProcManagement.html
1178 file path=usr/lib/help/profiles/locale/C/RtRepairMngmnt.html
1179 file path=usr/lib/help/profiles/locale/C/RtReservedProfile.html
1180 file path=usr/lib/help/profiles/locale/C/RtRightsDelegate.html
1181 file path=usr/lib/help/profiles/locale/C/RtSMBFSMngmnt.html
1182 file path=usr/lib/help/profiles/locale/C/RtSMBMngmnt.html
1183 file path=usr/lib/help/profiles/locale/C/RtSoftwareInstall.html

```
1184 file path=usr/lib/help/profiles/locale/C/RtSysAdmin.html
1185 file path=usr/lib/help/profiles/locale/C/RtSysEvMngmnt.html
1186 file path=usr/lib/help/profiles/locale/C/RtSysPowerMgmt.html
1187 file path=usr/lib/help/profiles/locale/C/RtSysPowerMgmtBrightness.html
1188 file path=usr/lib/help/profiles/locale/C/RtSysPowerMgmtSuspend.html
1189 file path=usr/lib/help/profiles/locale/C/RtSysPowerMgmtSuspendtoDisk.html
1190 file path=usr/lib/help/profiles/locale/C/RtSysPowerMgmtSuspendtoRAM.html
1191 file path=usr/lib/help/profiles/locale/C/RtUserMngmnt.html
1192 file path=usr/lib/help/profiles/locale/C/RtUserSecurity.html
1193 file path=usr/lib/help/profiles/locale/C/RtVscanMngmnt.html
1194 file path=usr/lib/help/profiles/locale/C/RtZFSFileSysMngmnt.html
1195 file path=usr/lib/help/profiles/locale/C/RtZFSStorageMngmnt.html
1196 file path=usr/lib/help/profiles/locale/C/RtZoneMngmnt.html
1197 file path=usr/lib/help/profiles/locale/C/RtZoneSecurity.html
1198 file path=usr/lib/hotplugd mode=0555
1199 file path=usr/lib/iconv/646da.8859.t mode=0444
1200 file path=usr/lib/iconv/646de.8859.t mode=0444
1201 file path=usr/lib/iconv/646en.8859.t mode=0444
1202 file path=usr/lib/iconv/646es.8859.t mode=0444
1203 file path=usr/lib/iconv/646fr.8859.t mode=0444
1204 file path=usr/lib/iconv/646it.8859.t mode=0444
1205 file path=usr/lib/iconv/646sv.8859.t mode=0444
1206 file path=usr/lib/iconv/8859.646.t mode=0444
1207 file path=usr/lib/iconv/8859.646da.t mode=0444
1208 file path=usr/lib/iconv/8859.646de.t mode=0444
1209 file path=usr/lib/iconv/8859.646en.t mode=0444
1210 file path=usr/lib/iconv/8859.646es.t mode=0444
1211 file path=usr/lib/iconv/8859.646fr.t mode=0444
1212 file path=usr/lib/iconv/8859.646it.t mode=0444
1213 file path=usr/lib/iconv/8859.646sv.t mode=0444
1214 file path=usr/lib/iconv/iconv_data mode=0444
1215 file path=usr/lib/imapd mode=0555
1216 file path=usr/lib/inet/$(ARCH32)/in.iked mode=0555
1217 file path=usr/lib/inet/$(ARCH64)/in.iked mode=0555
1218 file path=usr/lib/inet/certdb mode=0555
1219 file path=usr/lib/inet/certlocal mode=0555
1220 file path=usr/lib/inet/certrld mode=0555
1221 file path=usr/lib/inet/inetd mode=0555
1222 file path=usr/lib/intrd mode=0555
1222 file path=usr/lib/isaexec mode=0555
1223 file path=usr/lib/ksladm mode=0555
1224 file $(sparc_ONLY)file path=usr/lib/ld.so
1225 file path=usr/lib/libshare.so.1
1226 file path=usr/lib/makekey mode=0555
1227 file path=usr/lib/more.help
1228 file path=usr/lib/newsyslog group=sys mode=0555
1229 file path=usr/lib/passmgmt group=sys mode=0555
1230 file path=usr/lib/pci/pcidr mode=0555
1231 file path=usr/lib/pci/pcidr_plugin.so
1232 file path=usr/lib/pfexecd mode=0555
1233 file path=usr/lib/platexec mode=0555
1234 file path=usr/lib/rcm/modules/SUNW_aggr_rcm.so mode=0555
1235 file path=usr/lib/rcm/modules/SUNW_cluster_rcm.so mode=0555
1236 file path=usr/lib/rcm/modules/SUNW_dump_rcm.so mode=0555
1237 file path=usr/lib/rcm/modules/SUNW_filesys_rcm.so mode=0555
1238 file path=usr/lib/rcm/modules/SUNW_ibpart_rcm.so mode=0555
1239 file path=usr/lib/rcm/modules/SUNW_ip_anon_rcm.so mode=0555
1240 file path=usr/lib/rcm/modules/SUNW_ip_rcm.so mode=0555
1241 file path=usr/lib/rcm/modules/SUNW_mpxio_rcm.so mode=0555
1242 file path=usr/lib/rcm/modules/SUNW_network_rcm.so mode=0555
1243 file path=usr/lib/rcm/modules/SUNW_swap_rcm.so mode=0555
1244 file $(sparc_ONLY)file path=usr/lib/rcm/modules/SUNW_ttymux_rcm.so mode=0555
1245 file path=usr/lib/rcm/modules/SUNW_vlan_rcm.so mode=0555
1246 file path=usr/lib/rcm/modules/SUNW_vnic_rcm.so mode=0555
1247 file path=usr/lib/rcm/rcm_daemon mode=0555
1248 file path=usr/lib/reparse/reparsed group=sys mode=0555
```

```
1249 file path=usr/lib/saf/listen group=sys mode=0755
1250 file path=usr/lib/saf/nlps_server group=sys mode=0755
1251 file path=usr/lib/saf/sac group=sys mode=0555
1252 file path=usr/lib/saf/ttymon group=sys mode=0555
1253 file path=usr/lib/sysevent/modules/datalink_mod.so
1254 file path=usr/lib/sysevent/modules/devfsadmd_mod.so
1255 file path=usr/lib/sysevent/modules/sysevent_conf_mod.so
1256 file path=usr/lib/sysevent/modules/sysevent_reg_mod.so
1257 file path=usr/lib/sysevent/syseventconfd mode=0555
1258 file path=usr/lib/sysevent/syseventd mode=0555
1259 file path=usr/lib/utmp_update mode=4555
1260 file path=usr/lib/utmpd mode=0555
1261 file path=usr/lib/vtdaemon mode=0555
1262 file path=usr/lib/vtinfo mode=0555
1263 file path=usr/lib/vtxlock mode=0555
1264 file path=usr/sadm/bin/puttext mode=0555
1265 file path=usr/sadm/install/miniroot.db group=sys mode=0444
1266 file path=usr/sadm/install/scripts/i.ipsecalg group=sys mode=0555
1267 file path=usr/sadm/install/scripts/i.kcconf group=sys mode=0555
1268 file path=usr/sadm/install/scripts/i.kmconf group=sys mode=0555
1269 file path=usr/sadm/install/scripts/i.manifest group=sys mode=0555
1270 file path=usr/sadm/install/scripts/i.pkcs11conf group=sys mode=0555
1271 file path=usr/sadm/install/scripts/i.rbac group=sys mode=0555
1272 file path=usr/sadm/install/scripts/r.ipsecalg group=sys mode=0555
1273 file path=usr/sadm/install/scripts/r.kcconf group=sys mode=0555
1274 file path=usr/sadm/install/scripts/r.kmconf group=sys mode=0555
1275 file path=usr/sadm/install/scripts/r.manifest group=sys mode=0555
1276 file path=usr/sadm/install/scripts/r.pkcs11conf group=sys mode=0555
1277 file path=usr/sadm/install/scripts/r.rbac group=sys mode=0555
1278 file path=usr/sadm/updates mode=0444
1279 file $(i386_ONLY)file path=usr/sbin/$(ARCH32)/add_drv group=sys mode=0555
1280 file $(i386_ONLY)file path=usr/sbin/$(ARCH32)/modinfo group=sys mode=0555
1281 file $(i386_ONLY)file path=usr/sbin/$(ARCH32)/modload group=sys mode=0555
1282 file $(i386_ONLY)file path=usr/sbin/$(ARCH32)/modunload group=sys mode=0555
1283 file $(i386_ONLY)file path=usr/sbin/$(ARCH32)/pbind group=sys mode=0555
1284 file $(i386_ONLY)file path=usr/sbin/$(ARCH32)/prtconf group=sys mode=2555
1285 file $(i386_ONLY)file path=usr/sbin/$(ARCH32)/psrset group=sys mode=0555
1286 file $(i386_ONLY)file path=usr/sbin/$(ARCH32)/rem_drv group=sys mode=0555
1287 file $(i386_ONLY)file path=usr/sbin/$(ARCH32)/swap group=sys mode=2555
1288 file $(i386_ONLY)file path=usr/sbin/$(ARCH32)/sysdef group=sys mode=2555
1289 file $(i386_ONLY)file path=usr/sbin/$(ARCH32)/update_drv group=sys mode=0555
1290 file $(i386_ONLY)file path=usr/sbin/$(ARCH32)/whodo mode=4555
1291 file path=usr/sbin/$(ARCH64)/add_drv group=sys mode=0555
1292 file path=usr/sbin/$(ARCH64)/modinfo group=sys mode=0555
1293 file path=usr/sbin/$(ARCH64)/modload group=sys mode=0555
1294 file path=usr/sbin/$(ARCH64)/modunload group=sys mode=0555
1295 file path=usr/sbin/$(ARCH64)/pbind group=sys mode=0555
1296 file path=usr/sbin/$(ARCH64)/prtconf group=sys mode=2555
1297 file path=usr/sbin/$(ARCH64)/psrset group=sys mode=0555
1298 file path=usr/sbin/$(ARCH64)/rem_drv group=sys mode=0555
1299 file path=usr/sbin/$(ARCH64)/swap group=sys mode=2555
1300 file path=usr/sbin/$(ARCH64)/sysdef group=sys mode=2555
1301 file path=usr/sbin/$(ARCH64)/update_drv group=sys mode=0555
1302 file path=usr/sbin/$(ARCH64)/whodo mode=4555
1303 file path=usr/sbin/6to4relay mode=0555
1304 file path=usr/sbin/acctadm mode=0555
1305 file path=usr/sbin/allocate mode=4555
1306 file path=usr/sbin/arp mode=0555
1307 file path=usr/sbin/audit mode=0555
1308 file path=usr/sbin/auditconfig mode=0555
1309 file path=usr/sbin/auditd mode=0555
1310 file path=usr/sbin/auditrecord mode=0555
1311 file path=usr/sbin/auditreduce mode=0555
1312 file path=usr/sbin/auditstat mode=0555
1313 file path=usr/sbin/cfgadm mode=0555
1314 file path=usr/sbin/chroot mode=0555
```

new/usr/src/pkg/manifests/SUNWcs.mf

1315 file path=usr/sbin/clear_locks mode=0555
1316 file path=usr/sbin/clinfo mode=0555
1317 file path=usr/sbin/clri mode=0555
1318 file path=usr/sbin/consadm group=sys mode=0555
1319 file path=usr/sbin/cron group=sys mode=0555
1320 file path=usr/sbin/devfsadm group=sys mode=0755
1321 file path=usr/sbin/devinfo mode=0555
1322 file path=usr/sbin/df mode=0555
1323 file path=usr/sbin/dfmounts mode=0555
1324 file path=usr/sbin/dispadm mode=0555
1325 file path=usr/sbin/dminfo mode=0555
1326 file path=usr/sbin/dumpadm mode=0555
1327 file path=usr/sbin/EEPROM group=sys mode=2555
1328 file path=usr/sbin/ff mode=0555
1329 file path=usr/sbin/fmthard group=sys mode=0555
1330 file path=usr/sbin/format mode=0555
1331 file path=usr/sbin/fsck mode=0555
1332 file path=usr/sbin/fstyp group=sys mode=0555
1333 file path=usr/sbin/fuser mode=0555
1334 file path=usr/sbin/getdevpolicy group=sys mode=0555
1335 file path=usr/sbin/getmajor group=sys mode=0755
1336 file path=usr/sbin/groupadd group=sys mode=0555
1337 file path=usr/sbin/groupdel group=sys mode=0555
1338 file path=usr/sbin/groupmod group=sys mode=0555
1339 file path=usr/sbin/grpck mode=0555
1340 file path=usr/sbin/halt mode=0755
1341 file path=usr/sbin/hotplug mode=0555
1342 file path=usr/sbin/idmap mode=0555
1343 file path=usr/sbin/ifmpadm mode=0555
1344 file path=usr/sbin/ikeadm mode=0555
1345 file path=usr/sbin/ikecert mode=0555
1346 file path=usr/sbin/inetadm mode=0555
1347 file path=usr/sbin/inetconv mode=0555
1348 file path=usr/sbin/install mode=0555
1349 file path=usr/sbin/installboot group=sys mode=0555
1350 file path=usr/sbin/ipaddrsel mode=0555
1351 file path=usr/sbin/ipsecalg mode=0555
1352 file path=usr/sbin/ipsecconf mode=0555
1353 file path=usr/sbin/ipseckey mode=0555
1354 file path=usr/sbin/keyserv group=sys mode=0555
1355 file path=usr/sbin/killall mode=0555
1356 file path=usr/sbin/ksslcfg mode=0555
1357 file path=usr/sbin/link mode=0555
1358 file path=usr/sbin/locator mode=0555
1359 file path=usr/sbin/lofiadm mode=0555
1360 file path=usr/sbin/logadm mode=0555
1361 file path=usr/sbin/makedbm mode=0555
1362 file path=usr/sbin/mkdevalloc mode=0555
1363 file path=usr/sbin/mkfile mode=0555
1364 file path=usr/sbin/mknod mode=0555
1365 file path=usr/sbin/mountall group=sys mode=0555
1366 file path=usr/sbin/msgid mode=0555
1367 file path=usr/sbin/mvdir mode=0555
1368 file path=usr/sbin/ndd mode=0555
1369 file path=usr/sbin/nlsadmin group=adm mode=0755
1370 file path=usr/sbin/nscd mode=0555
1371 file path=usr/sbin/nwamadm mode=0555
1372 file path=usr/sbin/nwamcfg mode=0555
1373 file path=usr/sbin/pmadm group=sys mode=0555
1374 file path=usr/sbin/praudit mode=0555
1375 \$(i386_ONLY)file path=usr/sbin/prtdiag group=sys mode=2755
1376 file path=usr/sbin/prtvtoc group=sys mode=0555
1377 file path=usr/sbin/psradm group=sys mode=0555
1378 file path=usr/sbin/psrinfo group=sys mode=0555
1379 file path=usr/sbin/pwck mode=0555
1380 file path=usr/sbin/pwconv group=sys mode=0555

21

new/usr/src/pkg/manifests/SUNWcs.mf

1381 file path=usr/sbin/raidctl mode=0555
1382 file path=usr/sbin/ramdiskadm mode=0555
1383 file path=usr/sbin/rctldm mode=0555
1384 file path=usr/sbin/root_archive group=sys mode=0555
1385 file path=usr/sbin/rpcbind mode=0555
1386 \$(i386_ONLY)file path=usr/sbin/rtc mode=0555
1387 file path=usr/sbin/sacadm group=sys mode=4755
1388 file path=usr/sbin/setmnt mode=0555
1389 file path=usr/sbin/shareall mode=0555
1390 file path=usr/sbin/sharectl mode=0555
1391 file path=usr/sbin/sharemgr mode=0555
1392 file path=usr/sbin/shutdown group=sys mode=0755
1393 file path=usr/sbin/smbios mode=0555
1394 file path=usr/sbin/stmsboot mode=0555
1395 file path=usr/sbin/strace group=sys mode=0555
1396 file path=usr/sbin/strclean group=sys mode=0555
1397 file path=usr/sbin/strerr group=sys mode=0555
1398 file path=usr/sbin/sttydefs group=sys mode=0755
1399 file path=usr/sbin/svcadm mode=0555
1400 file path=usr/sbin/svccfg mode=0555
1401 file path=usr/sbin/syncinit mode=0555
1402 file path=usr/sbin/syncloop mode=0555
1403 file path=usr/sbin/syncstat mode=0555
1404 file path=usr/sbin/syseventadm group=sys mode=0555
1405 file path=usr/sbin/syslogd group=sys mode=0555
1406 file path=usr/sbin/tar mode=0555
1407 file path=usr/sbin/traceroute mode=4555
1408 file path=usr/sbin/trapstat mode=0555
1409 file path=usr/sbin/ttyadm group=sys mode=0755
1410 \$(i386_ONLY)file path=usr/sbin/ucodeadm mode=0555
1411 file path=usr/sbin/umountall group=sys mode=0555
1412 file path=usr/sbin/unlink mode=0555
1413 file path=usr/sbin/unshareall mode=0555
1414 file path=usr/sbin/useradd group=sys mode=0555
1415 file path=usr/sbin/userdel group=sys mode=0555
1416 file path=usr/sbin/usermod group=sys mode=0555
1417 \$(sparc_ONLY)file path=usr/sbin/virtinfo mode=0555
1418 file path=usr/sbin/volcopy mode=0555
1419 file path=usr/sbin/wall group=tty mode=2555
1420 file path=usr/sbin/zdump mode=0555
1421 file path=usr/sbin/zic mode=0555
1422 file path=usr/share/doc/ksh/COMPATIBILITY
1423 file path=usr/share/doc/ksh/DESIGN
1424 file path=usr/share/doc/ksh/OBSOLETE
1425 file path=usr/share/doc/ksh/README
1426 file path=usr/share/doc/ksh/RELEASE
1427 file path=usr/share/doc/ksh/TYPES
1428 file path=usr/share/doc/ksh/images/callouts/1.png
1429 file path=usr/share/doc/ksh/images/callouts/10.png
1430 file path=usr/share/doc/ksh/images/callouts/2.png
1431 file path=usr/share/doc/ksh/images/callouts/3.png
1432 file path=usr/share/doc/ksh/images/callouts/4.png
1433 file path=usr/share/doc/ksh/images/callouts/5.png
1434 file path=usr/share/doc/ksh/images/callouts/6.png
1435 file path=usr/share/doc/ksh/images/callouts/7.png
1436 file path=usr/share/doc/ksh/images/callouts/8.png
1437 file path=usr/share/doc/ksh/images/callouts/9.png
1438 file path=usr/share/doc/ksh/images/tag_bourne.png
1439 file path=usr/share/doc/ksh/images/tag_i18n.png
1440 file path=usr/share/doc/ksh/images/tag_ksh.png
1441 file path=usr/share/doc/ksh/images/tag_ksh88.png
1442 file path=usr/share/doc/ksh/images/tag_ksh93.png
1443 file path=usr/share/doc/ksh/images/tag_l10n.png
1444 file path=usr/share/doc/ksh/images/tag_perf.png
1445 file path=usr/share/doc/ksh/shell_styleguide.docbook
1446 file path=usr/share/doc/ksh/shell_styleguide.html

22

```
1447 file path=usr/share/lib/mailx/mailx.help
1448 file path=usr/share/lib/mailx/mailx.help.~
1449 file path=usr/share/lib/tabset/3101
1450 file path=usr/share/lib/tabset/beehive
1451 file path=usr/share/lib/tabset/hds
1452 file path=usr/share/lib/tabset/hds3
1453 file path=usr/share/lib/tabset/std
1454 file path=usr/share/lib/tabset/stdcrt
1455 file path=usr/share/lib/tabset/teleray
1456 file path=usr/share/lib/tabset/vt100
1457 file path=usr/share/lib/tabset/wyse-adds
1458 file path=usr/share/lib/tabset/xerox1720
1459 file path=usr/share/lib/termcap
1460 file path=usr/share/lib/unittab
1461 file path=usr/share/lib/xml/dtd/adt_record.dtd.1
1462 file path=usr/share/lib/xml/dtd/kmfpolicy.dtd
1463 file path=usr/share/lib/xml/dtd/service_bundle.dtd.1 group=sys
1464 file path=usr/share/lib/xml/style/adt_record.xsl.1
1465 file path=var/adm/aculog mode=0600 owner=uucp preserve=true
1466 file path=var/adm/spellhist mode=0666 preserve=true
1467 file path=var/adm/utmpx preserve=true
1468 file path=var/adm/wtmpx group=adm owner=adm preserve=true
1469 file path=var/log/authlog group=sys mode=0600 preserve=true
1470 file path=var/log/syslog group=sys preserve=true
1471 file path=var/sadm/system/admin/default_java group=sys mode=0444
1472 file path=var/saf/zsmon/log group=sys preserve=true
1473 file path=var/spool/cron/crontabs/adm group=sys mode=0600 preserve=true
1474 file path=var/spool/cron/crontabs/root group=sys mode=0600 preserve=true
1475 hardlink path=etc/rc2.d/S20syssetup target=../etc/init.d/syssetup
1476 hardlink path=etc/rc2.d/S73cachefs.daemon \
1477   target=../etc/init.d/cachefs.daemon
1478 hardlink path=etc/rc2.d/S89PRESERVE target=../etc/init.d/PRESERVE
1479 $(sparc_ONLY)hardlink path=etc/svc/profile/platform_SUNW,Sun-Fire-V890.xml \
1480   target=platform_SUNW,Sun-Fire-880.xml
1481 $(sparc_ONLY)hardlink \
1482   path=etc/svc/profile/platform_SUNW,UltraSPARC-IIe-NetraCT-40.xml \
1483   target=platform_SUNW,UltraSPARC-IIi-Netract.xml
1484 $(sparc_ONLY)hardlink \
1485   path=etc/svc/profile/platform_SUNW,UltraSPARC-IIe-NetraCT-60.xml \
1486   target=platform_SUNW,UltraSPARC-IIi-Netract.xml
1487 hardlink path=sbin/rc5 target=../sbin/rc0
1488 hardlink path=sbin/rc6 target=../sbin/rc0
1489 hardlink path=usr/bin/$(ARCH32)/encrypt target=decrypt
1490 hardlink path=usr/bin/$(ARCH32)/ksh target=ksh93
1491 hardlink path=usr/bin/$(ARCH32)/mac target=digest
1492 hardlink path=usr/bin/$(ARCH32)/rksh target=ksh93
1493 hardlink path=usr/bin/$(ARCH32)/rksh93 target=ksh93
1494 $(i386_ONLY)hardlink path=usr/bin/$(ARCH32)/w target=uptime
1495 hardlink path=usr/bin/$(ARCH64)/encrypt target=decrypt
1496 hardlink path=usr/bin/$(ARCH64)/ksh target=ksh93
1497 hardlink path=usr/bin/$(ARCH64)/mac target=digest
1498 hardlink path=usr/bin/$(ARCH64)/rksh target=ksh93
1499 hardlink path=usr/bin/$(ARCH64)/rksh93 target=ksh93
1500 hardlink path=usr/bin/$(ARCH64)/w target=uptime
1501 hardlink path=usr/bin/bg target=../usr/bin/alias
1502 hardlink path=usr/bin/cd target=../usr/bin/alias
1503 hardlink path=usr/bin/cksum target=../usr/bin/alias
1504 hardlink path=usr/bin/cmp target=../usr/bin/alias
1505 hardlink path=usr/bin/comm target=../usr/bin/alias
1506 hardlink path=usr/bin/command target=../usr/bin/alias
1507 hardlink path=usr/bin/cut target=../usr/bin/alias
1508 hardlink path=usr/bin/decrypt target=../usr/lib/isaexec
1509 hardlink path=usr/bin/digest target=../usr/lib/isaexec
1510 hardlink path=usr/bin/dispgid target=../usr/bin/ckgid
1511 hardlink path=usr/bin/dispuid target=../usr/bin/ckuid
1512 hardlink path=usr/bin/edit target=../has/bin/edit
```

```
1513 hardlink path=usr/bin/encrypt target=../usr/lib/isaexec
1514 hardlink path=usr/bin/fc target=../usr/bin/alias
1515 hardlink path=usr/bin/fg target=../usr/bin/alias
1516 hardlink path=usr/bin/getopts target=../usr/bin/alias
1517 hardlink path=usr/bin/hash target=../usr/bin/alias
1518 hardlink path=usr/bin/i386 target=../usr/bin/i286
1519 hardlink path=usr/bin/i486 target=../usr/bin/i286
1520 hardlink path=usr/bin/i860 target=../usr/bin/i286
1521 hardlink path=usr/bin/i86pc target=../usr/bin/i286
1522 hardlink path=usr/bin/iAPX286 target=../usr/bin/i286
1523 hardlink path=usr/bin/jobs target=../usr/bin/alias
1524 hardlink path=usr/bin/join target=../usr/bin/alias
1525 hardlink path=usr/bin/kill target=../usr/bin/alias
1526 hardlink path=usr/bin/ksh target=../usr/lib/isaexec
1527 hardlink path=usr/bin/ksh93 target=../usr/lib/isaexec
1528 hardlink path=usr/bin/ln target=../usr/bin/cp
1529 hardlink path=usr/bin/logname target=../usr/bin/alias
1530 hardlink path=usr/bin/m68k target=../usr/bin/i286
1531 hardlink path=usr/bin/mac target=../usr/lib/isaexec
1532 hardlink path=usr/bin/mc68000 target=../usr/bin/i286
1533 hardlink path=usr/bin/mc68010 target=../usr/bin/i286
1534 hardlink path=usr/bin/mc68020 target=../usr/bin/i286
1535 hardlink path=usr/bin/mc68030 target=../usr/bin/i286
1536 hardlink path=usr/bin/mc68040 target=../usr/bin/i286
1537 hardlink path=usr/bin/mv target=../usr/bin/cp
1538 hardlink path=usr/bin/newtask target=../usr/lib/isaexec
1539 hardlink path=usr/bin/nohup target=../usr/lib/isaexec
1540 hardlink path=usr/bin/page target=../usr/bin/more
1541 hardlink path=usr/bin/paste target=../usr/bin/alias
1542 hardlink path=usr/bin/pdp11 target=../usr/bin/i286
1543 hardlink path=usr/bin/pfbash target=../usr/bin/pfexec
1544 hardlink path=usr/bin/pfcsh target=../usr/bin/pfexec
1545 hardlink path=usr/bin/pfksh target=../usr/bin/pfexec
1546 hardlink path=usr/bin/pfksh93 target=../usr/bin/pfexec
1547 hardlink path=usr/bin/pfrksh target=../usr/bin/pfexec
1548 hardlink path=usr/bin/pfrksh93 target=../usr/bin/pfexec
1549 hardlink path=usr/bin/pfsh target=../usr/bin/pfexec
1550 hardlink path=usr/bin/pftcsh target=../usr/bin/pfexec
1551 hardlink path=usr/bin/pfzsh target=../usr/bin/pfexec
1552 hardlink path=usr/bin/pkill target=../usr/bin/pgrep
1553 hardlink path=usr/bin/prctl target=../usr/lib/isaexec
1554 hardlink path=usr/bin/print target=../usr/bin/alias
1555 hardlink path=usr/bin/prstat target=../usr/lib/isaexec
1556 hardlink path=usr/bin/ps target=../usr/lib/isaexec
1557 hardlink path=usr/bin/read target=../usr/bin/alias
1558 hardlink path=usr/bin/red target=../usr/bin/ed
1559 hardlink path=usr/bin/rev target=../usr/bin/alias
1560 hardlink path=usr/bin/rksh target=../usr/lib/isaexec
1561 hardlink path=usr/bin/rksh93 target=../usr/lib/isaexec
1562 hardlink path=usr/bin/savecore target=../usr/lib/isaexec
1563 hardlink path=usr/bin/setuname target=../usr/lib/isaexec
1564 hardlink path=usr/bin/sleep target=../usr/bin/alias
1565 hardlink path=usr/bin/sparc target=../usr/bin/i286
1566 hardlink path=usr/bin/sum target=../usr/bin/alias
1567 hardlink path=usr/bin/sun target=../usr/bin/i286
1568 hardlink path=usr/bin/sun2 target=../usr/bin/i286
1569 hardlink path=usr/bin/sun3 target=../usr/bin/i286
1570 hardlink path=usr/bin/sun3x target=../usr/bin/i286
1571 hardlink path=usr/bin/sun4 target=../usr/bin/i286
1572 hardlink path=usr/bin/sun4c target=../usr/bin/i286
1573 hardlink path=usr/bin/sun4d target=../usr/bin/i286
1574 hardlink path=usr/bin/sun4e target=../usr/bin/i286
1575 hardlink path=usr/bin/sun4m target=../usr/bin/i286
1576 hardlink path=usr/bin/tee target=../usr/bin/alias
1577 hardlink path=usr/bin/test target=../usr/bin/alias
1578 hardlink path=usr/bin/touch target=../usr/bin/settime
```



```

1579 hardlink path=usr/bin/type target=../usr/bin/alias
1580 hardlink path=usr/bin/u370 target=../usr/bin/i286
1581 hardlink path=usr/bin/u3b target=../usr/bin/i286
1582 hardlink path=usr/bin/u3b15 target=../usr/bin/i286
1583 hardlink path=usr/bin/u3b2 target=../usr/bin/i286
1584 hardlink path=usr/bin/u3b5 target=../usr/bin/i286
1585 hardlink path=usr/bin/ulimit target=../usr/bin/alias
1586 hardlink path=usr/bin/umask target=../usr/bin/alias
1587 hardlink path=usr/bin/unalias target=../usr/bin/alias
1588 hardlink path=usr/bin/uniq target=../usr/bin/alias
1589 hardlink path=usr/bin/uptime target=../usr/lib/isaexec
1590 hardlink path=usr/bin/vax target=../usr/bin/i286
1591 hardlink path=usr/bin/vedit target=../has/bin/edit
1592 hardlink path=usr/bin/w target=../usr/lib/isaexec
1593 hardlink path=usr/bin/wait target=../usr/bin/alias
1594 hardlink path=usr/bin/wc target=../usr/bin/alias
1595 hardlink path=usr/has/bin/ex target=edit
1596 hardlink path=usr/has/bin/pfsh target=../bin/pfexec
1597 hardlink path=usr/has/bin/vedit target=edit
1598 hardlink path=usr/has/bin/vi target=edit
1599 hardlink path=usr/has/bin/view target=edit
1600 hardlink path=usr/lib/fs/hsfs/fstyp target=../sbin/fstyp
1601 hardlink path=usr/lib/fs/ufs/dcopy target=../usr/lib/fs/ufs/clri
1602 hardlink path=usr/lib/fs/ufs/fstyp target=../sbin/fstyp
1603 hardlink path=usr/lib/fs/ufs/quotao \
1604     target=../usr/lib/fs/ufs/quotaooff
1605 hardlink path=usr/lib/inet/in.iked target=../usr/lib/isaexec
1606 hardlink path=usr/sadm/bin/dispgid target=../usr/bin/cgkid
1607 hardlink path=usr/sadm/bin/dispuuid target=../usr/bin/c kuid
1608 hardlink path=usr/sadm/bin/errange target=../usr/bin/ckrange
1609 hardlink path=usr/sadm/bin/errdate target=../usr/bin/ckdate
1610 hardlink path=usr/sadm/bin/errgid target=../usr/bin/cgkid
1611 hardlink path=usr/sadm/bin/errint target=../usr/bin/ckint
1612 hardlink path=usr/sadm/bin/erritem target=../usr/bin/ckitem
1613 hardlink path=usr/sadm/bin/errpath target=../usr/bin/ckpath
1614 hardlink path=usr/sadm/bin/errstr target=../usr/bin/ckstr
1615 hardlink path=usr/sadm/bin/errtime target=../usr/bin/cktime
1616 hardlink path=usr/sadm/bin/erruid target=../usr/bin/c kuid
1617 hardlink path=usr/sadm/bin/erryorn target=../usr/bin/ckyorn
1618 hardlink path=usr/sadm/bin/helpdate target=../usr/bin/ckdate
1619 hardlink path=usr/sadm/bin/helpgid target=../usr/bin/cgkid
1620 hardlink path=usr/sadm/bin/helpint target=../usr/bin/ckint
1621 hardlink path=usr/sadm/bin/helpitem target=../usr/bin/ckitem
1622 hardlink path=usr/sadm/bin/helppath target=../usr/bin/ckpath
1623 hardlink path=usr/sadm/bin/helpprange target=../usr/bin/ckrange
1624 hardlink path=usr/sadm/bin/helpstr target=../usr/bin/ckstr
1625 hardlink path=usr/sadm/bin/helptime target=../usr/bin/cktime
1626 hardlink path=usr/sadm/bin/helpuid target=../usr/bin/c kuid
1627 hardlink path=usr/sadm/bin/helpyorn target=../usr/bin/ckyorn
1628 hardlink path=usr/sadm/bin/valdate target=../usr/bin/ckdate
1629 hardlink path=usr/sadm/bin/valgid target=../usr/bin/cgkid
1630 hardlink path=usr/sadm/bin/valint target=../usr/bin/ckint
1631 hardlink path=usr/sadm/bin/valpath target=../usr/bin/ckpath
1632 hardlink path=usr/sadm/bin/valrange target=../usr/bin/ckrange
1633 hardlink path=usr/sadm/bin/valstr target=../usr/bin/ckstr
1634 hardlink path=usr/sadm/bin/valtime target=../usr/bin/cktime
1635 hardlink path=usr/sadm/bin/valuid target=../usr/bin/c kuid
1636 hardlink path=usr/sadm/bin/valyorn target=../usr/bin/ckyorn
1637 hardlink path=usr/sbin/add_drv target=../usr/lib/isaexec
1638 hardlink path=usr/sbin/audlinks target=../devfsadm
1639 hardlink path=usr/sbin/consadm target=../usr/sbin/consadm
1640 hardlink path=usr/sbin/deallocate target=../usr/sbin/allocate
1641 hardlink path=usr/sbin/devlinks target=../devfsadm
1642 hardlink path=usr/sbin/dfshares target=../usr/sbin/dfmounts
1643 hardlink path=usr/sbin/disks target=../devfsadm
1644 hardlink path=usr/sbin/drvconfig target=../devfsadm

```

```

1645 hardlink path=usr/sbin/list_devices target=../usr/sbin/allocate
1646 hardlink path=usr/sbin/mkdevmaps target=../usr/sbin/mkdevalloc
1647 hardlink path=usr/sbin/modinfo target=../usr/lib/isaexec
1648 hardlink path=usr/sbin/modload target=../usr/lib/isaexec
1649 hardlink path=usr/sbin/modunload target=../usr/lib/isaexec
1650 hardlink path=usr/sbin/pbind target=../usr/lib/isaexec
1651 hardlink path=usr/sbin/ports target=../devfsadm
1652 hardlink path=usr/sbin/poweroff target=../halt
1653 hardlink path=usr/sbin/prtconf target=../usr/lib/isaexec
1654 $(sparc ONLY)hardlink path=usr/sbin/prtdiag target=../usr/lib/latexex
1655 hardlink path=usr/sbin/psrset target=../usr/lib/isaexec
1656 hardlink path=usr/sbin/reboot target=../halt
1657 hardlink path=usr/sbin/rem_drv target=../usr/lib/isaexec
1658 hardlink path=usr/sbin/roleadd target=../usr/sbin/useradd
1659 hardlink path=usr/sbin/roledel target=../usr/sbin/userdel
1660 hardlink path=usr/sbin/rolemod target=../usr/sbin/usermod
1661 hardlink path=usr/sbin/share target=../usr/sbin/sharemgr
1662 hardlink path=usr/sbin/swap target=../usr/lib/isaexec
1663 hardlink path=usr/sbin/sysdef target=../usr/lib/isaexec
1664 hardlink path=usr/sbin/tapes target=../devfsadm
1665 hardlink path=usr/sbin/unshare target=../usr/sbin/sharemgr
1666 hardlink path=usr/sbin/update_drv target=../usr/lib/isaexec
1667 hardlink path=usr/sbin/whodo target=../usr/lib/isaexec
1668 legacy pkg=SUNWcsr \
1669     desc="core software for a specific instruction-set architecture" \
1670     name="Core Solaris, (Root)"
1671 legacy pkg=SUNWcsu \
1672     desc="core software for a specific instruction-set architecture" \
1673     name="Core Solaris, (Usr)"
1674 legacy pkg=SUNWftpr desc="FTP Server Configuration Files" \
1675     name="FTP Server, (Root)"
1676 license cr_Sun license=cr_Sun
1677 license lic_CDDL license=lic_CDDL
1678 license usr/src/cmd/cmd-inet/sbin/ifparse/THIRDPARTYLICENSE \
1679     license=usr/src/cmd/cmd-inet/sbin/ifparse/THIRDPARTYLICENSE
1680 license usr/src/cmd/cmd-inet/usr.lib/in.mpathd/THIRDPARTYLICENSE \
1681     license=usr/src/cmd/cmd-inet/usr.lib/in.mpathd/THIRDPARTYLICENSE
1682 license usr/src/cmd/cmd-inet/usr.sbin/THIRDPARTYLICENSE.arp \
1683     license=usr/src/cmd/cmd-inet/usr.sbin/THIRDPARTYLICENSE.arp
1684 license usr/src/cmd/cmd-inet/usr.sbin/THIRDPARTYLICENSE.route \
1685     license=usr/src/cmd/cmd-inet/usr.sbin/THIRDPARTYLICENSE.route
1686 license usr/src/cmd/cmd-inet/usr.sbin/ifconfig/THIRDPARTYLICENSE \
1687     license=usr/src/cmd/cmd-inet/usr.sbin/ifconfig/THIRDPARTYLICENSE
1688 license usr/src/cmd/cmd-inet/usr.sbin/in.ftpd/LICENSE \
1689     license=usr/src/cmd/cmd-inet/usr.sbin/in.ftpd/LICENSE
1690 license usr/src/cmd/cmd-inet/usr.sbin/traceroute/THIRDPARTYLICENSE \
1691     license=usr/src/cmd/cmd-inet/usr.sbin/traceroute/THIRDPARTYLICENSE
1692 license usr/src/cmd/cron/THIRDPARTYLICENSE \
1693     license=usr/src/cmd/cron/THIRDPARTYLICENSE
1694 license usr/src/cmd/csh/THIRDPARTYLICENSE \
1695     license=usr/src/cmd/csh/THIRDPARTYLICENSE
1696 license usr/src/cmd/eeprom/THIRDPARTYLICENSE \
1697     license=usr/src/cmd/eeprom/THIRDPARTYLICENSE
1698 license usr/src/cmd/fs.d/ufs/THIRDPARTYLICENSE \
1699     license=usr/src/cmd/fs.d/ufs/THIRDPARTYLICENSE
1700 license usr/src/cmd/mt/THIRDPARTYLICENSE \
1701     license=usr/src/cmd/mt/THIRDPARTYLICENSE
1702 license usr/src/cmd/script/THIRDPARTYLICENSE \
1703     license=usr/src/cmd/script/THIRDPARTYLICENSE
1704 license usr/src/cmd/sed/THIRDPARTYLICENSE \
1705     license=usr/src/cmd/sed/THIRDPARTYLICENSE
1706 license usr/src/cmd/stat/vmstat/THIRDPARTYLICENSE \
1707     license=usr/src/cmd/stat/vmstat/THIRDPARTYLICENSE
1708 license usr/src/cmd/tail/THIRDPARTYLICENSE \
1709     license=usr/src/cmd/tail/THIRDPARTYLICENSE
1710 license usr/src/cmd/tip/THIRDPARTYLICENSE \

```

```

1711     license=usr/src/cmd/tip/THIRDPARTYLICENSE
1712 license usr/src/cmd/tr/THIRDPARTYLICENSE \
1713     license=usr/src/cmd/tr/THIRDPARTYLICENSE
1714 license usr/src/cmd/vi/THIRDPARTYLICENSE \
1715     license=usr/src/cmd/vi/THIRDPARTYLICENSE
1716 license usr/src/cmd/which/THIRDPARTYLICENSE \
1717     license=usr/src/cmd/which/THIRDPARTYLICENSE
1718 license usr/src/cmd/xstr/THIRDPARTYLICENSE \
1719     license=usr/src/cmd/xstr/THIRDPARTYLICENSE
1720 license usr/src/common/bzip2/LICENSE license=usr/src/common/bzip2/LICENSE
1721 link path=bin target=./usr/bin
1722 link path=etc/TIMEZONE target=./default/init
1723 link path=etc/autopush target=./sbin/autopush
1724 link path=etc/cfgadm target=./usr/sbin/cfgadm
1725 link path=etc/clri target=./usr/sbin/clri
1726 link path=etc/cron target=./usr/sbin/cron
1727 link path=etc/dcopy target=./usr/sbin/dcopy
1728 link path=etc/ff target=./usr/sbin/ff
1729 link path=etc/fmthard target=./usr/sbin/fmthard
1730 link path=etc/format target=./usr/sbin/format
1731 link path=etc/fsck target=./usr/sbin/fsck
1732 link path=etc/fsdb target=./usr/sbin/fsdb
1733 link path=etc/fstyp target=./usr/sbin/fstyp
1734 link path=etc/getty target=./usr/lib/saf/ttymon
1735 link path=etc/grpck target=./usr/sbin/grpck
1736 link path=etc/halt target=./usr/sbin/halt
1737 link path=etc/hosts target=./inet/hosts
1738 link path=etc/inet/ipnodes target=./hosts
1739 link path=etc/inetd.conf target=./inet/inetd.conf
1740 link path=etc/init target=./sbin/init
1741 link path=etc/install target=./usr/sbin/install
1742 link path=etc/killall target=./usr/sbin/killall
1743 link path=etc/labelit target=./usr/sbin/labelit
1744 link path=etc/lib/ld.so.1 target=./../lib/ld.so.1
1745 link path=etc/lib/libdl.so.1 target=./../lib/libdl.so.1
1746 link path=etc/lib/nss_files.so.1 target=./../lib/nss_files.so.1
1747 link path=etc/log target=./var/adm/log
1748 link path=etc/mkfs target=./usr/sbin/mkfs
1749 link path=etc/mknod target=./usr/sbin/mknod
1750 link path=etc/mount target=./sbin/mount
1751 link path=etc/mountall target=./sbin/mountall
1752 link path=etc/ncheck target=./usr/sbin/ncheck
1753 link path=etc/netmasks target=./inet/netmasks
1754 link path=etc/networks target=./inet/networks
1755 link path=etc/protocols target=./inet/protocols
1756 link path=etc/prtconf target=./usr/sbin/prtconf
1757 link path=etc/prtvtoc target=./usr/sbin/prtvtoc
1758 link path=etc/rc0 target=./sbin/rc0
1759 link path=etc/rc1 target=./sbin/rc1
1760 link path=etc/rc2 target=./sbin/rc2
1761 link path=etc/rc3 target=./sbin/rc3
1762 link path=etc/rc5 target=./sbin/rc5
1763 link path=etc/rc6 target=./sbin/rc6
1764 link path=etc/rcS target=./sbin/rcS
1765 link path=etc/reboot target=./usr/sbin/halt
1766 link path=etc/security/audit/localhost/files target=./../../var/audit
1767 link path=etc/services target=./inet/services
1768 link path=etc/setmnt target=./usr/sbin/setmnt
1769 link path=etc/shutdown target=./usr/sbin/shutdown
1770 link path=etc/sulogin target=./sbin/sulogin
1771 link path=etc/swap target=./usr/sbin/swap
1772 link path=etc/swapadd target=./sbin/swapadd
1773 link path=etc/sysdef target=./usr/sbin/sysdef
1774 link path=etc/tar target=./usr/sbin/tar
1775 link path=etc/telinit target=./sbin/init
1776 link path=etc/uadmin target=./sbin/uadmin

```

```

1777 link path=etc/umount target=./sbin/umount
1778 link path=etc/umountall target=./sbin/umountall
1779 link path=etc/utmpx target=./var/adm/utmpx
1780 link path=etc/volcopy target=./usr/sbin/volcopy
1781 link path=etc/wall target=./usr/sbin/wall
1782 link path=etc/whodo target=./usr/sbin/whodo
1783 link path=etc/wtmpx target=./var/adm/wtmpx
1784 link path=sbin/in.mpathd target=./lib/inet/in.mpathd
1785 link path=sbin/jsh target=./usr/bin/ksh93
1786 link path=sbin/pfsh target=./usr/bin/pfexec
1787 link path=sbin/sh target=./usr/bin/$(ARCH32)/ksh93
1788 link path=sbin/su target=./usr/bin/su
1789 link path=usr/adm target=./var/adm
1790 link path=usr/bin/cachefspack target=./lib/fs/cachefs/cachefspack
1791 link path=usr/bin/cachefsstat target=./lib/fs/cachefs/cachefsstat
1792 link path=usr/bin/df target=./sbin/df
1793 link path=usr/bin/jsh target=ksh93
1794 link path=usr/bin/pwconv target=./sbin/pwconv
1795 link path=usr/bin/rmail target=./mail
1796 link path=usr/bin/sh target=$(ARCH32)/ksh93
1797 link path=usr/bin/strclean target=./sbin/strclean
1798 link path=usr/bin/strerr target=./sbin/strerr
1799 link path=usr/bin/sync target=./../sbin/sync
1800 link path=usr/bin/tar target=./sbin/tar
1801 link path=usr/bin/uname target=./../sbin/uname
1802 link path=usr/ccs/bin/m4 target=./../bin/m4
1803 link path=usr/has/bin/jsh target=sh
1804 link path=usr/has/lib/rsh target=./bin/sh
1805 link path=usr/lib/$(ARCH64)/ld.so.1 target=./../lib/$(ARCH64)/ld.so.1
1806 link path=usr/lib/cron target=./../etc/cron.d
1807 link path=usr/lib/devfsadm/devfsadmd target=./sbin/devfsadm
1808 link path=usr/lib/embedded_su target=./bin/su
1809 link path=usr/lib/fs/dev/mount target=./../../etc/fs/dev/mount
1810 link path=usr/lib/fs/hsfs/mount target=./../../etc/fs/hsfs/mount
1811 link path=usr/lib/fs/ufs/mount target=./../../etc/fs/ufs/mount
1812 link path=usr/lib/inet/in.mpathd target=./../lib/inet/in.mpathd
1813 link path=usr/lib/ld.so.1 target=./../lib/ld.so.1
1814 link path=usr/lib/locale/POSIX target=./C
1815 link path=usr/lib/rsh target=./bin/ksh93
1816 link path=usr/lib/secure/32 target=.
1817 link path=usr/lib/secure/64 target=$(ARCH64)
1818 link path=usr/lib/wusbd target=./../sbin/wusbadm
1819 link path=usr/mail target=./var/mail
1820 link path=usr/net/nls/listen target=./../lib/saf/listen
1821 link path=usr/net/nls/nlps_server target=./../lib/saf/nlps_server
1822 link path=usr/news target=./var/news
1823 link path=usr/preserve target=./var/preserve
1824 link path=usr/pub target=./share/lib/pub
1825 link path=usr/sbin/autopush target=./../sbin/autopush
1826 link path=usr/sbin/bootadm target=./../sbin/bootadm
1827 link path=usr/sbin/cachefslog target=./lib/fs/cachefs/cachefslog
1828 link path=usr/sbin/cachefswssize target=./lib/fs/cachefs/cachefswssize
1829 link path=usr/sbin/cfsadmin target=./lib/fs/cachefs/cfsadmin
1830 link path=usr/sbin/cryptoadm target=./../sbin/cryptoadm
1831 link path=usr/sbin/dcopy target=./clri
1832 link path=usr/sbin/devnm target=./df
1833 link path=usr/sbin/dladm target=./../sbin/dladm
1834 link path=usr/sbin/dlstat target=./../sbin/dlstat
1835 link path=usr/sbin/edquota target=./lib/fs/ufs/edquota
1836 link path=usr/sbin/fdisk target=./../sbin/fdisk
1837 link path=usr/sbin/fiocompress target=./../sbin/fiocompress
1838 link path=usr/sbin/flowadm target=./../sbin/flowadm
1839 link path=usr/sbin/flowstat target=./../sbin/flowstat
1840 link path=usr/sbin/fsdb target=./clri
1841 link path=usr/sbin/fsirand target=./lib/fs/ufs/fsirand
1842 link path=usr/sbin/fssnap target=./clri

```

```
1843 link path=usr/sbin/hostconfig target=../sbin/hostconfig
1844 link path=usr/sbin/ifconfig target=../sbin/ifconfig
1845 link path=usr/sbin/inetd target=../lib/inet/inetd
1846 link path=usr/sbin/init target=../sbin/init
1847 $(i386_ONLY)link path=usr/sbin/installgrub target=../sbin/installgrub
1848 link path=usr/sbin/ipadm target=../sbin/ipadm
1849 link path=usr/sbin/impstat target=../sbin/impstat
1850 link path=usr/sbin/labelit target=../cli
1851 link path=usr/sbin/lockfs target=../lib/fs/ufs/lockfs
1852 link path=usr/sbin/mkfs target=../cli
1853 link path=usr/sbin/mount target=../sbin/mount
1854 link path=usr/sbin/ncheck target=../ff
1855 link path=usr/sbin/newfs target=../lib/fs/ufs/newfs
1856 link path=usr/sbin/quot target=../lib/fs/ufs/quot
1857 link path=usr/sbin/quota target=../lib/fs/ufs/quota
1858 link path=usr/sbin/quotacheck target=../lib/fs/ufs/quotacheck
1859 link path=usr/sbin/quotaoff target=../lib/fs/ufs/quotaoff
1860 link path=usr/sbin/quotaon target=../lib/fs/ufs/quotaon
1861 link path=usr/sbin/repquota target=../lib/fs/ufs/repquota
1862 link path=usr/sbin/route target=../sbin/route
1863 link path=usr/sbin/routed target=../sbin/routed
1864 link path=usr/sbin/sync target=../sbin/sync
1865 link path=usr/sbin/tunefs target=../lib/fs/ufs/tunefs
1866 link path=usr/sbin/tzreload target=../sbin/tzreload
1867 link path=usr/sbin/uadmin target=../sbin/uadmin
1868 link path=usr/sbin/ufsdump target=../lib/fs/ufs/ufsdump
1869 link path=usr/sbin/ufsrestore target=../lib/fs/ufs/ufsrestore
1870 link path=usr/sbin/umount target=../sbin/umount
1871 link path=usr/sbin/wusbadm target=../sbin/wusbadm
1872 link path=usr/spool target=../var/spool
1873 link path=usr/src target=../share/src
1874 link path=usr/tmp target=../var/tmp
1875 link path=var/ld/32 target=
1876 link path=var/ld/64 target=$(ARCH64)
1877 #
1878 # The bootadm binary needs the etc/release file.
1879 #
1880 depend fmri=release/name type=require
1881 #
1882 # illumos-defaulted perl interpreter
1883 # intrd and others use the illumos-defaulted perl interpreter
1884 depend fmri=runtime/perl-510 type=require
1885 #
1886 # The loadkeys binary needs the keytables.
1887 #
1888 depend fmri=system/data/keyboard/keytables type=require
1889 #
1890 # Depend on terminfo data.
1891 #
1892 depend fmri=system/data/terminfo type=require
1893 #
1894 # Depend on zoneinfo data.
1895 #
1896 depend fmri=system/data/zoneinfo type=require
```

new/usr/src/uts/common/Makefile.files

1

43688 Sat Sep 14 15:00:29 2013

new/usr/src/uts/common/Makefile.files

XXX Rearchitect and replace interrupt distribution

```
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2012 Nexenta Systems, Inc. All rights reserved.
25 # Copyright (c) 2012 by Delphix. All rights reserved.
26 # Copyright (c) 2013 by Saso Kiselkov. All rights reserved.
27 #
28 #
29 #
30 # This Makefile defines all file modules for the directory uts/common
31 # and its children. These are the source files which may be considered
32 # common to all SunOS systems.
33 #
34 i386_CORE_OBJS += \
35     atomic.o      \
36     avintr.o      \
37     pic.o
38 #
39 sparc_CORE_OBJS +=
40 #
41 COMMON_CORE_OBJS += \
42     beep.o        \
43     bitset.o      \
44     bp_map.o      \
45     brand.o       \
46     cpucaps.o     \
47     cmt.o         \
48     cmt_policy.o  \
49     cpu.o         \
50     cpu_event.o   \
51     cpu_intr.o    \
52     cpu_pm.o      \
53     cpupart.o     \
54     cap_util.o    \
55     disp.o        \
56     group.o       \
57     kstat_fr.o    \
58     iscsiboot_prop.o \
59     lgrp.o         \
60     lgrp_topo.o   \
61     mmapobj.o     \
```

new/usr/src/uts/common/Makefile.files

2

```
62     mutex.o       \
63     page_lock.o   \
64     page_retire.o \
65     panic.o       \
66     param.o       \
67     pg.o          \
68     pghw.o        \
69     putnext.o     \
70     rctl_proc.o   \
71     rwlock.o      \
72     seg_kmem.o    \
73     softint.o     \
74     string.o      \
75     strtol.o      \
76     strtoul.o     \
77     strtoll.o     \
78     strtoull.o    \
79     thread_intr.o \
80     vm_page.o     \
81     vm_pagelist.o \
82     zlib_obj.o    \
83     clock_tick.o
84 #
85 CORE_OBJS += $(COMMON_CORE_OBJS) $(MACH)_CORE_OBJS
86 #
87 ZLIB_OBJS = \
88     zutil.o \
89     zmod.o \
90     zmod_subr.o \
91     adler32.o \
92     crc32.o \
93     deflate.o \
94     inflate.o \
95     inf_trees.o \
96     trees.o
97 #
98 GENUNIX_OBJS += \
99     access.o \
100     acl.o \
101     acl_common.o \
102     adjtime.o \
103     alarm.o \
104     aio_subr.o \
105     auditsys.o \
106     audit_core.o \
107     audit_zone.o \
108     audit_memory.o \
109     autoconf.o \
110     avl.o \
111     bdev_dsort.o \
112     bio.o \
113     bitmap.o \
114     blabel.o \
115     brandsys.o \
116     bz2blocksort.o \
117     bz2compress.o \
118     bz2decompress.o \
119     bz2randtable.o \
120     bz2bzip.o \
121     bz2crctable.o \
122     bz2huffman.o \
123     callb.o \
124     callout.o \
125     chdir.o \
126     chmod.o \
127     chown.o \
128     cladm.o \
129     class.o \
130     clock.o \
131     clock_highres.o \
132     clock_realtime.o \
133     close.o \
134     compress.o \
```

new/usr/src/uts/common/Makefile.files

```

128      condvar.o \
129      conf.o \
130      console.o \
131      contract.o \
132      copyops.o \
133      core.o \
134      corectl.o \
135      cred.o \
136      cs_stubs.o \
137      dacf.o \
138      dacf_clnt.o \
139      damap.o \
140      cyclic.o \
141      ddi.o \
142      ddifm.o \
143      ddi_hp_impl.o \
144      ddi_hp_ndi.o \
145      ddi_intr.o \
146      ddi_intr_impl.o \
147      ddi_intr_irm.o \
148      ddi_nodeid.o \
149      ddi_periodic.o \
150      devcfg.o \
151      devcache.o \
152      device.o \
153      devid.o \
154      devid_cache.o \
155      devid_scsi.o \
156      devid_smp.o \
157      devpolicy.o \
158      disp_lock.o \
159      dnld.o \
160      driver.o \
161      dumpsubr.o \
162      driver_lyr.o \
163      dtrace_subr.o \
164      errorq.o \
165      etheraddr.o \
166      evchannels.o \
167      exacct.o \
168      exacct_core.o \
169      exec.o \
170      exit.o \
171      fbio.o \
172      fcctl.o \
173      fdbuffer.o \
174      fdsync.o \
175      fem.o \
176      ffs.o \
177      fio.o \
178      flock.o \
179      fm.o \
180      fork.o \
181      vpm.o \
182      fs_reparse.o \
183      fs_subr.o \
184      fsflush.o \
185      ftrace.o \
186      getcwd.o \
187      getdents.o \
188      getloadavg.o \
189      getpagesizes.o \
190      getpid.o \
191      gfs.o \
192      rusagesys.o \
193      gid.o \

```

3

new/usr/src/uts/common/Makefile.files

```

194      groups.o \
195      grow.o \
196      hat_refmod.o \
197      interrupt_balancer.o \
198      #endif /* ! codereview */
199      id32.o \
200      id_space.o \
201      inet_ntop.o \
202      instance.o \
203      ioctl.o \
204      ip_cksum.o \
205      issetugid.o \
206      ippconf.o \
207      kpcp.o \
208      kdi.o \
209      kiconv.o \
210      klpd.o \
211      kmem.o \
212      ksyms_snapshot.o \
213      l_strplumb.o \
214      labelsys.o \
215      link.o \
216      list.o \
217      lockstat_subr.o \
218      log_sysevent.o \
219      logsubr.o \
220      lookup.o \
221      lseek.o \
222      ltos.o \
223      lwp.o \
224      lwp_create.o \
225      lwp_info.o \
226      lwp_self.o \
227      lwp_sobj.o \
228      lwp_timer.o \
229      lwpsys.o \
230      main.o \
231      mmapobjsys.o \
232      memcntl.o \
233      memstr.o \
234      mgrpsys.o \
235      mkdir.o \
236      mknod.o \
237      mount.o \
238      move.o \
239      msacct.o \
240      multidata.o \
241      nbmlock.o \
242      ndifm.o \
243      nice.o \
244      netstack.o \
245      ntptime.o \
246      nvpair.o \
247      nvpair_alloc_system.o \
248      nvpair_alloc_fixed.o \
249      fnvpair.o \
250      octet.o \
251      open.o \
252      p_online.o \
253      pathconf.o \
254      pathname.o \
255      pause.o \
256      serializer.o \
257      pci_intr_lib.o \
258      pci_cap.o \
259      pcifm.o \

```

4

```

260      pgrp.o          \
261      pgrpsys.o       \
262      pid.o           \
263      pkp_hash.o      \
264      policy.o        \
265      poll.o          \
266      pool.o          \
267      pool_pset.o     \
268      port_subr.o     \
269      ppriv.o         \
270      printf.o        \
271      priocntl.o      \
272      priv.o          \
273      priv_const.o    \
274      proc.o          \
275      procset.o       \
276      processor_bind.o \
277      processor_info.o \
278      profil.o        \
279      project.o       \
280      qsort.o         \
281      rctl.o          \
282      rctlsys.o       \
283      readlink.o      \
284      refstr.o        \
285      rename.o        \
286      resolvepath.o  \
287      retire_store.o  \
288      process.o       \
289      rlimit.o        \
290      rmap.o          \
291      rw.o            \
292      rwstlock.o      \
293      sad_conf.o      \
294      sid.o           \
295      sidsys.o        \
296      sched.o         \
297      schedctl.o      \
298      sctp_crc32.o    \
299      seg_dev.o       \
300      seg_kp.o        \
301      seg_kpm.o       \
302      seg_map.o       \
303      seg_vn.o        \
304      seg_spt.o       \
305      semaphore.o     \
306      sendfile.o      \
307      session.o       \
308      share.o         \
309      shuttle.o       \
310      sig.o           \
311      sigaction.o     \
312      sigaltstack.o   \
313      signotify.o      \
314      sigpending.o    \
315      sigprocmask.o   \
316      sigqueue.o      \
317      sigsendset.o    \
318      sigsuspend.o    \
319      sigtimedwait.o  \
320      sleepq.o        \
321      sock_conf.o     \
322      space.o         \
323      sscanf.o        \
324      stat.o          \
325      statfs.o        \

```

```

326      statvfs.o       \
327      stol.o          \
328      str_conf.o      \
329      strcalls.o      \
330      stream.o        \
331      streamio.o      \
332      strext.o        \
333      strsubr.o       \
334      strsun.o        \
335      subr.o          \
336      sunddi.o        \
337      sunmdi.o        \
338      sunndi.o        \
339      sunpci.o        \
340      sunpm.o         \
341      sundlpi.o       \
342      suntpi.o        \
343      swap_subr.o     \
344      swap_vnops.o    \
345      symlink.o       \
346      sync.o          \
347      sysclass.o      \
348      sysconfig.o     \
349      sysent.o        \
350      sysfs.o         \
351      systeminfo.o    \
352      task.o          \
353      taskq.o         \
354      tasksys.o       \
355      time.o          \
356      timer.o         \
357      times.o         \
358      timers.o        \
359      thread.o        \
360      tlabel.o        \
361      tn timer.o       \
362      turnstile.o     \
363      tty_common.o    \
364      u8_textprep.o   \
365      uadmin.o        \
366      uconv.o         \
367      ucredsys.o      \
368      uid.o           \
369      umask.o         \
370      umount.o        \
371      uname.o         \
372      unix_bb.o       \
373      unlink.o        \
374      urw.o           \
375      utime.o         \
376      utssys.o        \
377      uucopy.o        \
378      vfs.o           \
379      vfs_conf.o      \
380      vmem.o          \
381      vm_anon.o       \
382      vm_as.o         \
383      vm_meter.o      \
384      vm_pageout.o    \
385      vm_pvn.o        \
386      vm_rm.o         \
387      vm_seg.o        \
388      vm_subr.o       \
389      vm_swap.o       \
390      vm_usage.o      \
391      vnode.o         \

```

```

392         vuid_queue.o \
393         vuid_store.o \
394         waitq.o \
395         watchpoint.o \
396         yield.o \
397         scsi_confdata.o \
398         xattr.o \
399         xattr_common.o \
400         xdr_mblk.o \
401         xdr_mem.o \
402         xdr.o \
403         xdr_array.o \
404         xdr_refer.o \
405         xhat.o \
406         zone.o

408 #
409 #       Stubs for the stand-alone linker/loader
410 #
411 sparc_GENSTUBS_OBJS = \
412         kobj_stubs.o

414 i386_GENSTUBS_OBJS =

416 COMMON_GENSTUBS_OBJS =

418 GENSTUBS_OBJS += $(COMMON_GENSTUBS_OBJS) ${$(MACH)_GENSTUBS_OBJS}

420 #
421 #       DTrace and DTrace Providers
422 #
423 DTRACE_OBJS += dtrace.o dtrace_isa.o dtrace_asm.o

425 SDT_OBJS += sdt_subr.o

427 PROFILE_OBJS += profile.o

429 SYSTRACE_OBJS += systrace.o

431 LOCKSTAT_OBJS += lockstat.o

433 FASTTRAP_OBJS += fasttrap.o fasttrap_isa.o

435 DCPC_OBJS += dcpc.o

437 #
438 #       Driver (pseudo-driver) Modules
439 #
440 IPP_OBJS += ippctl.o

442 AUDIO_OBJS += audio_client.o audio_ddi.o audio_engine.o \
443         audio_fltdata.o audio_format.o audio_ctrl.o \
444         audio_grc3.o audio_output.o audio_input.o \
445         audio_oss.o audio_sun.o

447 AUDIOEMU10K_OBJS += audioemu10k.o

449 AUDIOENS_OBJS += audioens.o

451 AUDIOVIA823X_OBJS += audiovia823x.o

453 AUDIOVIA97_OBJS += audiovia97.o

455 AUDIO1575_OBJS += audio1575.o

457 AUDIO810_OBJS += audio810.o

```

```

459 AUDIOCMI_OBJS += audiocmi.o

461 AUDIOCMIHD_OBJS += audiocmihd.o

463 AUDIOHD_OBJS += audiohd.o

465 AUDIOIXP_OBJS += audioixp.o

467 AUDIOLS_OBJS += audiols.o

469 AUDIOP16X_OBJS += audiop16x.o

471 AUDIOPCI_OBJS += audiopci.o

473 AUDIOSOLO_OBJS += audiosolo.o

475 AUDIOTS_OBJS += audiotics.o

477 AC97_OBJS += ac97.o ac97_ad.o ac97_alc.o ac97_cmi.o

479 BLKDEV_OBJS += blkdev.o

481 CARDBUS_OBJS += cardbus.o cardbus_hp.o cardbus_cfg.o

483 CONSKBD_OBJS += conskbd.o

485 CONSMS_OBJS += consms.o

487 OLDPTY_OBJS += tty_ptyconf.o

489 PTC_OBJS += tty_pty.o

491 PTSL_OBJS += tty_pts.o

493 PTM_OBJS += ptm.o

495 MII_OBJS += mii.o mii_cicada.o mii_natsemi.o mii_intel.o mii_qualsemi.o \
496         mii_marvell.o mii_realtek.o mii_other.o

498 PTS_OBJS += pts.o

500 PTY_OBJS += ptms_conf.o

502 SAD_OBJS += sad.o

504 MD4_OBJS += md4.o md4_mod.o

506 MD5_OBJS += md5.o md5_mod.o

508 SHA1_OBJS += sha1.o sha1_mod.o

510 SHA2_OBJS += sha2.o sha2_mod.o

512 IPGPC_OBJS += classifierddi.o classifier.o filters.o trie.o table.o \
513         ba_table.o

515 DSCPMK_OBJS += dscpmk.o dscpmkddi.o

517 DLCOSMK_OBJS += dlcosmk.o dlcosmkddi.o

519 FLOWACCT_OBJS += flowacctddi.o flowacct.o

521 TOKENMT_OBJS += tokenmt.o tokenmtddi.o

523 TSWTCL_OBJS += tswtcl.o tswtclddi.o

```

```

525 ARP_OBJS +=      arpddi.o
527 ICMP_OBJS +=     icmpddi.o
529 ICMP6_OBJS +=    icmp6ddi.o
531 RTS_OBJS +=       rtsddi.o

533 IP_ICMP_OBJS =     icmp.o icmp_opt_data.o
534 IP_RTS_OBJS =       rts.o rts_opt_data.o
535 IP_TCP_OBJS =        tcp.o tcp_fusion.o tcp_opt_data.o tcp_sack.o tcp_stats.o \
536 tcp_misc.o tcp_timers.o tcp_time_wait.o tcp_tpi.o tcp_output.o \
537 tcp_input.o tcp_socket.o tcp_bind.o tcp_cluster.o tcp_tunables.o
538 IP_UDP_OBJS =        udp.o udp_opt_data.o udp_tunables.o udp_stats.o
539 IP_SCTP_OBJS =       sctp.o sctp_opt_data.o sctp_output.o \
540 sctp_init.o sctp_input.o sctp_cookie.o \
541 sctp_conn.o sctp_error.o sctp_snmp.o \
542 sctp_tunables.o sctp_shutdown.o sctp_common.o \
543 sctp_timer.o sctp_heartbeat.o sctp_hash.o \
544 sctp_bind.o sctp_notify.o sctp_asconf.o \
545 sctp_addr.o tn_ipopt.o tnet.o ip_netinfo.o \
546 sctp_misc.o
547 IP_ILB_OBJS =        ilb.o ilb_nat.o ilb_conn.o ilb_alg_hash.o ilb_alg_rr.o

549 IP_OBJS +=          igmp.o ipmp.o ip.o ip6.o ip6_asp.o ip6_if.o ip6_ire.o \
550 ip6_rts.o ip_if.o ip_ire.o ip_listutils.o ip_mroute.o \
551 ip_multi.o ip2mac.o ip_ndp.o ip_rts.o ip_srcid.o \
552 ipddi.o ipdrop.o mi.o nd.o tunables.o optcom.o snmpcom.o \
553 ipsec_loader.o spd.o ipclassifier.o inet_common.o ip_queue.o \
554 queue.o ip_sadb.o ip_ftable.o proto_set.o radix.o ip_dummy.o \
555 ip_helper_stream.o ip_tunables.o \
556 ip_output.o ip_input.o ip6_input.o ip6_output.o ip_arp.o \
557 conn_opt.o ip_attr.o ip_dce.o \
558 $(IP_ICMP_OBJS) \
559 $(IP_RTS_OBJS) \
560 $(IP_TCP_OBJS) \
561 $(IP_UDP_OBJS) \
562 $(IP_SCTP_OBJS) \
563 $(IP_ILB_OBJS)

565 IP6_OBJS +=         ip6ddi.o
567 HOOK_OBJS +=        hook.o
569 NETI_OBJS +=         neti_impl.o neti_mod.o neti_stack.o

571 KEYSOCK_OBJS +=     keysockddi.o keysock.o keysock_opt_data.o
573 IPNET_OBJS +=       ipnet.o ipnet_bpf.o

575 SPDSOCK_OBJS +=     spdsockddi.o spdsock.o spdsock_opt_data.o
577 IPSECESP_OBJS +=    ipsecespddi.o ipsecesp.o
579 IPSECAH_OBJS +=     ipsecahddi.o ipsecah.o sadb.o

581 SPPP_OBJS +=         sPPP.o sPPP_dlp_i.o sPPP_mod.o s_common.o
583 SPPPTUN_OBJS +=      sppptun.o sppptun_mod.o
585 SPPPASYN_OBJS +=     spppasyn.o spppasyn_mod.o
587 SPPPCOMP_OBJS +=     sPPPcomp.o sPPPcomp_mod.o deflate.o bsd-comp.o vjcompress.o \
588 zlib.o

```

```

590 TCP_OBJS +=         tcpddi.o
592 TCP6_OBJS +=        tcp6ddi.o
594 NCA_OBJS +=          ncaddi.o
596 SDP SOCK_MOD_OBJS += sockmod_sdp.o socksdp.o socksdpsubr.o
598 SCTP SOCK_MOD_OBJS += sockmod_sctp.o socksctp.o socksctpsubr.o
600 PFP SOCK_MOD_OBJS += sockmod_pfp.o
602 RDS SOCK_MOD_OBJS += sockmod_rds.o
604 RDS_OBJS +=          rdsddi.o rdssubr.o rds_opt.o rds_ioctl.o
606 RDSIB_OBJS +=         rdsib.o rdsib_ib.o rdsib_cm.o rdsib_ep.o rdsib_buf.o \
607 rdsib_debug.o rdsib_sc.o
609 RDSV3_OBJS +=         af_rds.o rds_v3_ddi.o bind.o loop.o threads.o connection.o \
610 transport.o cong.o sysctl.o message.o rds_rcv.o send.o \
611 stats.o info.o page.o rdma_transport.o ib_ring.o ib_rdma.o \
612 ib_rcv.o ib.o ib_send.o ib_sysctl.o ib_stats.o ib_cm.o \
613 rds_v3_sc.o rds_v3_debug.o rds_v3_impl.o rdma.o rds_v3_af_thr.o

615 ISER_OBJS +=          iser.o iser_cm.o iser_cq.o iser_ib.o iser_idm.o \
616 iser_resource.o iser_xfer.o
618 UDP_OBJS +=           udpddi.o
620 UDP6_OBJS +=          udp6ddi.o
622 SY_OBJS +=            genty.o
624 TCO_OBJS +=           ticots.o
626 TCOO_OBJS +=          ticotsord.o
628 TCL_OBJS +=           ticlts.o
630 TL_OBJS +=            tl.o
632 DUMP_OBJS +=          dump.o
634 BPF_OBJS +=           bpf.o bpf_filter.o bpf_mod.o bpf_dlt.o bpf_mac.o
636 CLONE_OBJS +=         clone.o
638 CN_OBJS +=            cons.o
640 DLD_OBJS +=           dld_drv.o dld_proto.o dld_str.o dld_flow.o
642 DLS_OBJS +=           dls.o dls_link.o dls_mod.o dls_stat.o dls_mgmt.o
644 GLD_OBJS +=           gld.o gldutil.o
646 MAC_OBJS +=           mac.o mac_bcast.o mac_client.o mac_datapath_setup.o mac_flow.o
647 mac_hio.o mac_mod.o mac_ndd.o mac_provider.o mac_sched.o \
648 mac_protect.o mac_soft_ring.o mac_stat.o mac_util.o

650 MAC_6TO4_OBJS +=      mac_6to4.o
652 MAC_ETHER_OBJS +=     mac_ether.o
654 MAC_IPV4_OBJS +=      mac_ipv4.o

```



```
656 MAC_IPV6_OBJS +=      mac_ipv6.o
658 MAC_WIFI_OBJS +=      mac_wifi.o
660 MAC_IB_OBJS +=         mac_ib.o
662 IPTUN_OBJS +=         iptun_dev.o iptun_ctl.o iptun.o
664 AGGR_OBJS +=          aggr_dev.o aggr_ctl.o aggr_grp.o aggr_port.o \
665                        aggr_send.o aggr_recv.o aggr_lacp.o
667 SOFTMAC_OBJS +=        softmac_main.o softmac_ctl.o softmac_capab.o \
668                        softmac_dev.o softmac_stat.o softmac_pkt.o softmac_fp.o
670 NET80211_OBJS +=       net80211.o net80211_proto.o net80211_input.o \
671                        net80211_output.o net80211_node.o net80211_crypto.o \
672                        net80211_crypto_none.o net80211_crypto_wep.o net80211_ioctl.o \
673                        net80211_crypto_tkip.o net80211_crypto_ccmp.o \
674                        net80211_ht.o
676 VNIC_OBJS +=          vnictl.o vnict_dev.o
678 SIMNET_OBJS +=        simnet.o
680 IB_OBJS +=            ibnex.o ibnex_ioctl.o ibnex_hca.o
682 IBCM_OBJS +=          ibcm_impl.o ibcm_sm.o ibcm_ti.o ibcm_utils.o ibcm_path.o \
683                        ibcm_arp.o ibcm_arp_link.o
685 IBDM_OBJS +=          ibdm.o
687 IBDMA_OBJS +=         ibdma.o
689 IBMF_OBJS +=          ibmf.o ibmf_impl.o ibmf_dr.o ibmf_wqe.o ibmf_ud_dest.o ibmf_mod.o
690                        ibmf_send.o ibmf_recv.o ibmf_handlers.o ibmf_trans.o \
691                        ibmf_timers.o ibmf_msg.o ibmf_utils.o ibmf_rmpp.o \
692                        ibmf_saa.o ibmf_saa_impl.o ibmf_saa_utils.o ibmf_saa_events.o
694 IBTL_OBJS +=          ibtl_impl.o ibtl_util.o ibtl_mem.o ibtl_handlers.o ibtl_qp.o \
695                        ibtl_cq.o ibtl_wr.o ibtl_hca.o ibtl_chan.o ibtl_cm.o \
696                        ibtl_mcg.o ibtl_ibnex.o ibtl_srq.o ibtl_part.o
698 TAVOR_OBJS +=         tavor.o tavor_agents.o tavor_cfg.o tavor_ci.o tavor_cmd.o \
699                        tavor_cq.o tavor_event.o tavor_ioctl.o tavor_misc.o \
700                        tavor_mr.o tavor_qp.o tavor_qpm.o tavor_rsrc.o \
701                        tavor_srq.o tavor_stats.o tavor_umap.o tavor_wr.o
703 HERMON_OBJS +=        hermon.o hermon_agents.o hermon_cfg.o hermon_ci.o hermon_cmd.o \
704                        hermon_cq.o hermon_event.o hermon_ioctl.o hermon_misc.o \
705                        hermon_mr.o hermon_qp.o hermon_qpm.o hermon_rsrc.o \
706                        hermon_srq.o hermon_stats.o hermon_umap.o hermon_wr.o \
707                        hermon_fcoib.o hermon_fm.o
709 DAPLT_OBJS +=         daplt.o
711 SOL_OFS_OBJS +=       sol_cma.o sol_ib_cma.o sol_uobj.o \
712                        sol_ofs_debug_util.o sol_ofs_gen_util.o \
713                        sol_kverbs.o
715 SOL_UCMA_OBJS +=      sol_ucma.o
717 SOL_UVERBS_OBJS +=    sol_uverbs.o sol_uverbs_comp.o sol_uverbs_event.o \
718                        sol_uverbs_hca.o sol_uverbs_qp.o
720 SOL_UMAD_OBJS +=      sol_umad.o
```

```
722 KSTAT_OBJS +=        kstat.o
724 KSYMS_OBJS +=         ksyms.o
726 INSTANCE_OBJS +=      inst_sync.o
728 IWSCN_OBJS +=         iwscons.o
730 LOFI_OBJS +=          lofi.o LzmaDec.o
732 FSSNAP_OBJS +=        fssnap.o
734 FSSNAPIF_OBJS +=       fssnap_if.o
736 MM_OBJS +=            mem.o
738 PHYSMEM_OBJS +=       physmem.o
740 OPTIONS_OBJS +=       options.o
742 WINLOCK_OBJS +=       winlockio.o
744 PM_OBJS +=            pm.o
745 SRN_OBJS +=            srn.o
747 PSEUDO_OBJS +=        pseudonex.o
749 RAMDISK_OBJS +=       ramdisk.o
751 LLC1_OBJS +=          llc1.o
753 USBKBM_OBJS +=        usbkbm.o
755 USBWCM_OBJS +=        usbwcm.o
757 BOFI_OBJS +=          bofi.o
759 HID_OBJS +=           hid.o
761 HWA_RC_OBJS +=        hwarc.o
763 USBSKEL_OBJS +=        usbkskel.o
765 USBVC_OBJS +=          usbvc.o usbvc_v4l2.o
767 HIDPARSER_OBJS +=      hidparser.o
769 USB_AC_OBJS +=         usb_ac.o
771 USB_AS_OBJS +=         usb_as.o
773 USB_AH_OBJS +=         usb_ah.o
775 USBMS_OBJS +=          usbms.o
777 USBPRN_OBJS +=         usbprn.o
779 UGEN_OBJS +=           ugen.o
781 USBSER_OBJS +=         usbser.o usbser_rseq.o
783 USBSACM_OBJS +=        usb_sacm.o
785 USBSER_KEYSPAN_OBJS += usbser_keyspan.o keyspan_dsd.o keyspan_pipe.o
787 USBS49_FW_OBJS +=      keyspan_49fw.o
```

```

789 USBSPRL_OBJS += usbser_pl2303.o pl2303_dsd.o
791 WUSB_CA_OBJS += wusb_ca.o
793 USBFTDI_OBJS += usbser_uftdi.o uftdi_dsd.o
795 USBECM_OBJS += usbecm.o
797 WC_OBJS += wscons.o vcons.o
799 VCONS_CONF_OBJS += vcons_conf.o
801 SCSI_OBJS +=      scsi_capabilities.o scsi_confsubr.o scsi_control.o \
802                scsi_data.o scsi_fm.o scsi_hba.o scsi_reset_notify.o \
803                scsi_resource.o scsi_subr.o scsi_transport.o scsi_watch.o \
804                smp_transport.o
806 SCSI_VHCI_OBJS +=      scsi_vhci.o mpapi_impl.o scsi_vhci_tpgs.o
808 SCSI_VHCI_F_SYM_OBJS +=      sym.o
810 SCSI_VHCI_F_TPGS_OBJS +=      tpgs.o
812 SCSI_VHCI_F_ASYM_SUN_OBJS +=  asym_sun.o
814 SCSI_VHCI_F_SYM_HDS_OBJS +=  sym_hds.o
816 SCSI_VHCI_F_TAPE_OBJS +=      tape.o
818 SCSI_VHCI_F_TPGS_TAPE_OBJS +=  tpgs_tape.o
820 SGEN_OBJS +=      sgen.o
822 SMP_OBJS +=      smp.o
824 SATA_OBJS +=      sata.o
826 USBA_OBJS +=      hcdi.o usba.o usbai.o hubdi.o parser.o genconsole.o \
827                usbai_pipe_mgmt.o usbai_req.o usbai_util.o usbai_register.o \
828                usba_devdb.o usba10_calls.o usba_uugen.o whcdi.o wa.o
829 USBA_WITHOUT_WUSB_OBJS +=      hcdi.o usba.o usbai.o hubdi.o parser.o gencons
830                usbai_pipe_mgmt.o usbai_req.o usbai_util.o usbai_register.o \
831                usba_devdb.o usba10_calls.o usba_uugen.o
833 USBA10_OBJS +=      usba10.o
835 RSM_OBJS +=      rsm.o rsmka_pathmanager.o rsmka_util.o
837 RSMOPS_OBJS +=      rsmops.o
839 S1394_OBJS +=      t1394.o t1394_errmsg.o s1394.o s1394_addr.o s1394_async.o \
840                s1394_bus_reset.o s1394_cmp.o s1394_csr.o s1394_dev_disc.o \
841                s1394_fa.o s1394_fcp.o \
842                s1394_hotplug.o s1394_isoch.o s1394_misc.o h1394.o nx1394.o
844 HCI1394_OBJS +=      hci1394.o hci1394_async.o hci1394_attach.o hci1394_buf.o \
845                hci1394_csr.o hci1394_detach.o hci1394_extern.o \
846                hci1394_ioctl.o hci1394_isoch.o hci1394_isr.o \
847                hci1394_ixl_comp.o hci1394_ixl_isr.o hci1394_ixl_misc.o \
848                hci1394_ixl_update.o hci1394_misc.o hci1394_ohci.o \
849                hci1394_q.o hci1394_s1394if.o hci1394_tlabel.o \
850                hci1394_tlist.o hci1394_vendor.o
852 AV1394_OBJS +=      av1394.o av1394_as.o av1394_async.o av1394_cfgrom.o \
853                av1394_cmp.o av1394_fcp.o av1394_isoch.o av1394_isoch_chan.o \

```

```

854                av1394_isoch_recv.o av1394_isoch_xmit.o av1394_list.o \
855                av1394_queue.o
857 DCAM1394_OBJS +=      dcam.o dcam_frame.o dcam_param.o dcam_reg.o \
858                dcam_ring_buff.o
860 SCSA1394_OBJS +=      hba.o sbp2_driver.o sbp2_bus.o
862 SBP2_OBJS +=      cfgrom.o sbp2.o
864 PMODEM_OBJS +=      pmodem.o pmodem_cis.o cis.o cis_callout.o cis_handlers.o cis_para
866 DSW_OBJS +=      dsw.o dsw_dev.o ii_tree.o
868 NCALL_OBJS +=      ncall.o \
869                ncall_stub.o
871 RDC_OBJS +=      rdc.o \
872                rdc_dev.o \
873                rdc_io.o \
874                rdc_clnt.o \
875                rdc_prot_xdr.o \
876                rdc_svc.o \
877                rdc_bitmap.o \
878                rdc_health.o \
879                rdc_subr.o \
880                rdc_diskq.o
882 RDCSRV_OBJS +=      rdcsrv.o
884 RDCSTUB_OBJS +=      rdc_stub.o
886 SDBC_OBJS +=      sd_bcache.o \
887                sd_bio.o \
888                sd_conf.o \
889                sd_ft.o \
890                sd_hash.o \
891                sd_io.o \
892                sd_misc.o \
893                sd_pcu.o \
894                sd_tdaemon.o \
895                sd_trace.o \
896                sd_iob_impl0.o \
897                sd_iob_impl1.o \
898                sd_iob_impl2.o \
899                sd_iob_impl3.o \
900                sd_iob_impl4.o \
901                sd_iob_impl5.o \
902                sd_iob_impl6.o \
903                sd_iob_impl7.o \
904                safestore.o \
905                safestore_ram.o
907 NSCTL_OBJS +=      nsctl.o \
908                nsc_cache.o \
909                nsc_disk.o \
910                nsc_dev.o \
911                nsc_freeze.o \
912                nsc_gen.o \
913                nsc_mem.o \
914                nsc_ncallio.o \
915                nsc_power.o \
916                nsc_resv.o \
917                nsc_rmspin.o \
918                nsc_solaris.o \
919                nsc_trap.o \

```

```
new/usr/src/uts/common/Makefile.files
```

```

986 TNF_OBJS +=      tnfnbuf.o      tnfntrace.o      tnfnwriter.o      trace_init.o \
987      trace_funcs.o      tnfnprobe.o      tnfn.o

989 LOGINDMUX_OBJS += logindmux.o

991 DEVINFO_OBJS += devinfo.o

993 DEVPOLL_OBJS += devpoll.o

995 DEVPOOL_OBJS += devpool.o

997 I8042_OBJS +=      i8042.o

999 KB8042_OBJS +=      \
1000      at_keyprocess.o \
1001      kb8042.o      \
1002      kb8042_keytables.o

1004 MOUSE8042_OBJS += mouse8042.o

1006 FDC_OBJS +=      fdc.o

1008 ASY_OBJS +=      asy.o

1010 ECPP_OBJS +=      ecpp.o

1012 VUIDM3P_OBJS += vuidmice.o vuidm3p.o

1014 VUIDM4P_OBJS += vuidmice.o vuidm4p.o

1016 VUIDM5P_OBJS += vuidmice.o vuidm5p.o

1018 VUIDPS2_OBJS += vuidmice.o vuidps2.o

1020 HPCSVCS_OBJS += hpcsvc.o

1022 PCIE_MISC_OBJS += pcie.o pcie_fault.o pcie_hp.o pciehpc.o pcishpc.o pcie_pwr.o p
1024 PCIHPNEXUS_OBJS += pcihp.o

1026 OPENEEPR_OBJS += openprom.o

1028 RANDOM_OBJS += random.o

1030 PSHOT_OBJS += pshot.o

1032 GEN_DRV_OBJS += gen_drv.o

1034 TCLIENT_OBJS += tclient.o

1036 TPHCI_OBJS += tphci.o

1038 TVHCI_OBJS += tvhci.o

1040 EMUL64_OBJS += emul64.o emul64_bsd.o

1042 FCP_OBJS += fcp.o

1044 FCIP_OBJS += fcip.o

1046 FCSM_OBJS += fcsm.o

1048 FCTL_OBJS += fctl.o

1050 FP_OBJS += fp.o

```

```

1052 QLC_OBJS += ql_api.o ql_debug.o ql_hba_fru.o ql_init.o ql_iocb.o ql_ioctl.o \
1053         ql_isr.o ql_mbx.o ql_nx.o ql_xioctl.o ql_fw_table.o

1055 QLC_FW_2200_OBJS += ql_fw_2200.o

1057 QLC_FW_2300_OBJS += ql_fw_2300.o

1059 QLC_FW_2400_OBJS += ql_fw_2400.o

1061 QLC_FW_2500_OBJS += ql_fw_2500.o

1063 QLC_FW_6322_OBJS += ql_fw_6322.o

1065 QLC_FW_8100_OBJS += ql_fw_8100.o

1067 QLGE_OBJS += qlge.o qlge_dbg.o qlge_flash.o qlge_fm.o qlge_gld.o qlge_mpi.o

1069 ZCONS_OBJS += zcons.o

1071 NV_SATA_OBJS += nv_sata.o

1073 SI3124_OBJS += si3124.o

1075 AHCI_OBJS += ahci.o

1077 PCIIDE_OBJS += pci-ide.o

1079 PCEPP_OBJS += pcepp.o

1081 CPC_OBJS += cpc.o

1083 CPUID_OBJS += cpuid_drv.o

1085 SYSEVENT_OBJS += sysevent.o

1087 BL_OBJS += bl.o

1089 DRM_OBJS += drm_sunmod.o drm_kstat.o drm_agpsupport.o \
1090         drm_auth.o drm_bufs.o drm_context.o drm_dma.o \
1091         drm_drawable.o drm_drv.o drm_fops.o drm_ioctl.o drm_irq.o \
1092         drm_lock.o drm_memory.o drm_msg.o drm_pci.o drm_scatter.o \
1093         drm_cache.o drm_gem.o drm_mm.o ati_pcigart.o

1095 FM_OBJS += devfm.o devfm_machdep.o

1097 RTLS_OBJS += rtls.o

1099 #
1100 #             exec modules
1101 #
1102 AOUTEXEC_OBJS +=aout.o

1104 ELFEXEC_OBJS += elf.o elf_notes.o old_notes.o

1106 INTPEXEC_OBJS +=intp.o

1108 SHBINEXEC_OBJS +=shbin.o

1110 JAVAEXEC_OBJS +=java.o

1112 #
1113 #             file system modules
1114 #
1115 AUTOFS_OBJS += auto_vfsops.o auto_vnops.o auto_subr.o auto_xdr.o auto_sys.o

1117 CACHEFS_OBJS += cachefs_cnode.o         cachefs_cod.o \

```

```

1118         cachefs_dir.o         cachefs_dlog.o cachefs_filegrp.o \
1119         cachefs_fscache.o     cachefs_ioctl.o cachefs_log.o \
1120         cachefs_module.o \
1121         cachefs_noopc.o       cachefs_resource.o \
1122         cachefs_strict.o \
1123         cachefs_subr.o        cachefs_vfsops.o \
1124         cachefs_vnops.o

1126 DCFS_OBJS += dc_vnops.o

1128 DEVFS_OBJS += devfs_subr.o devfs_vfsops.o devfs_vnops.o

1130 DEV_OBJS += sdev_subr.o sdev_vfsops.o sdev_vnops.o \
1131         sdev_ptsops.o sdev_zvolops.o sdev_comm.o \
1132         sdev_profile.o sdev_ncache.o sdev_netops.o \
1133         sdev_ipnetops.o \
1134         sdev_vtops.o

1136 CTFS_OBJS += ctfs_all.o ctfs_cdir.o ctfs_ctl.o ctfs_event.o \
1137         ctfs_latest.o ctfs_root.o ctfs_sym.o ctfs_tdir.o ctfs_tmpl.o

1139 OBJFS_OBJS += objfs_vfs.o objfs_root.o objfs_common.o \
1140         objfs_odir.o objfs_data.o

1142 FDFS_OBJS += fdops.o

1144 FIFO_OBJS += fifosubr.o fifovnops.o

1146 PIPE_OBJS += pipe.o

1148 HSFS_OBJS += hsfs_node.o hsfs_subr.o hsfs_vfsops.o hsfs_vnops.o \
1149         hsfs_susp.o hsfs_rrip.o hsfs_susp_subr.o

1151 LOFS_OBJS += lofs_subr.o lofs_vfsops.o lofs_vnops.o

1153 NAMEFS_OBJS += namevfs.o namevno.o

1155 NFS_OBJS += nfs_client.o nfs_common.o nfs_dump.o \
1156         nfs_subr.o nfs_vfsops.o nfs_vnops.o \
1157         nfs_xdr.o nfs_sys.o nfs_strerror.o \
1158         nfs3_vfsops.o nfs3_vnops.o nfs3_xdr.o \
1159         nfs_acl_vnops.o nfs_acl_xdr.o nfs4_vfsops.o \
1160         nfs4_vnops.o nfs4_xdr.o nfs4_idmap.o \
1161         nfs4_shadow.o nfs4_subr.o \
1162         nfs4_attr.o nfs4_rnode.o nfs4_client.o \
1163         nfs4_acache.o nfs4_common.o nfs4_client_state.o \
1164         nfs4_callback.o nfs4_recovery.o nfs4_client_secinfo.o \
1165         nfs4_client_debug.o nfs_stats.o \
1166         nfs4_acl.o nfs4_stub_vnops.o nfs_cmd.o

1168 NFSSRV_OBJS += nfs_server.o nfs_srv.o nfs3_srv.o \
1169         nfs_acl_srv.o nfs_auth.o nfs_auth_xdr.o \
1170         nfs_export.o nfs_log.o nfs_log_xdr.o \
1171         nfs4_srv.o nfs4_state.o nfs4_srv_attr.o \
1172         nfs4_srv_ns.o nfs4_db.o nfs4_srv_deleg.o \
1173         nfs4_deleg_ops.o nfs4_srv_readdir.o nfs4_dispatch.o

1175 SMBSRV_SHARED_OBJS += \
1176         smb_inet.o \
1177         smb_match.o \
1178         smb_msgbuf.o \
1179         smb_oem.o \
1180         smb_string.o \
1181         smb_utf8.o \
1182         smb_door_legacy.o \
1183         smb_xdr.o \

```

```

1184      smb_token.o \
1185      smb_token_xdr.o \
1186      smb_sid.o \
1187      smb_native.o \
1188      smb_netbios_util.o

1190  SMBSRV_OBJS += $(SMBSRV_SHARED_OBJS) \
1191      smb_acl.o \
1192      smb_alloc.o \
1193      smb_close.o \
1194      smb_common_open.o \
1195      smb_common_transact.o \
1196      smb_create.o \
1197      smb_delete.o \
1198      smb_directory.o \
1199      smb_dispatch.o \
1200      smb_echo.o \
1201      smb_fem.o \
1202      smb_find.o \
1203      smb_flush.o \
1204      smb_fsinfo.o \
1205      smb_fsops.o \
1206      smb_init.o \
1207      smb_kdoor.o \
1208      smb_kshare.o \
1209      smb_kutil.o \
1210      smb_lock.o \
1211      smb_lock_byte_range.o \
1212      smb_locking_andx.o \
1213      smb_logoff_andx.o \
1214      smb_mangle_name.o \
1215      smb_mbuf_marshaling.o \
1216      smb_mbuf_util.o \
1217      smb_negotiate.o \
1218      smb_net.o \
1219      smb_node.o \
1220      smb_nt_cancel.o \
1221      smb_nt_create_andx.o \
1222      smb_nt_transact_create.o \
1223      smb_nt_transact_ioctl.o \
1224      smb_nt_transact_notify_change.o \
1225      smb_nt_transact_quota.o \
1226      smb_nt_transact_security.o \
1227      smb_odir.o \
1228      smb_ofile.o \
1229      smb_open_andx.o \
1230      smb_opipe.o \
1231      smb_oplock.o \
1232      smb_pathname.o \
1233      smb_print.o \
1234      smb_process_exit.o \
1235      smb_query_fileinfo.o \
1236      smb_read.o \
1237      smb_rename.o \
1238      smb_sd.o \
1239      smb_seek.o \
1240      smb_server.o \
1241      smb_session.o \
1242      smb_session_setup_andx.o \
1243      smb_set_fileinfo.o \
1244      smb_signing.o \
1245      smb_tree.o \
1246      smb_trans2_create_directory.o \
1247      smb_trans2_dfs.o \
1248      smb_trans2_find.o \
1249      smb_tree_connect.o

```

```

1250      smb_unlock_byte_range.o \
1251      smb_user.o \
1252      smb_vfs.o \
1253      smb_vops.o \
1254      smb_vss.o \
1255      smb_write.o \
1256      smb_write_raw.o

1258  PCFS_OBJS += pc_alloc.o pc_dir.o pc_node.o pc_subr.o \
1259      pc_vfsops.o pc_vnops.o

1261  PROC_OBJS += prcontrol.o priocntl.o prsubr.o prusrrio.o \
1262      prvfops.o prvnops.o

1264  MNTFS_OBJS += mntvfops.o mntvnops.o

1266  SHAREFS_OBJS += sharetab.o sharefs_vfsops.o sharefs_vnops.o

1268  SPEC_OBJS += specsubr.o specvfops.o specvnops.o

1270  SOCK_OBJS += socksubr.o sockvfops.o sockparams.o \
1271      socksyscalls.o socktpi.o sockstr.o \
1272      sockcommon_vnops.o sockcommon_subr.o \
1273      sockcommon_sops.o sockcommon.o \
1274      sock_notsupp.o socknotify.o \
1275      nl7c.o nl7curi.o nl7http.o nl7clogd.o \
1276      nl7cnca.o sodirect.o sockfilter.o

1278  TMPFS_OBJS += tmp_dir.o tmp_subr.o tmp_tnode.o tmp_vfsops.o \
1279      tmp_vnops.o

1281  UDFS_OBJS += udf_alloc.o udf_bmap.o udf_dir.o \
1282      udf_inode.o udf_subr.o udf_vfsops.o \
1283      udf_vnops.o

1285  UFS_OBJS += ufs_alloc.o ufs_bmap.o ufs_dir.o ufs_xattr.o \
1286      ufs_inode.o ufs_subr.o ufs_tables.o ufs_vfsops.o \
1287      ufs_vnops.o quota.o quotacalls.o quota_ufs.o \
1288      ufs_filio.o ufs_lockfs.o ufs_thread.o ufs_trans.o \
1289      ufs_acl.o ufs_panic.o ufs_directio.o ufs_log.o \
1290      ufs_extvnops.o ufs_snap.o lufs.o lufs_thread.o \
1291      lufs_log.o lufs_map.o lufs_top.o lufs_debug.o \
1292      vscan_drv.o vscan_svc.o vscan_door.o

1294  NSMB_OBJS += smb_conn.o smb_dev.o smb_iod.o smb_pass.o \
1295      smb_rq.o smb_sign.o smb_smb.o smb_subrs.o \
1296      smb_time.o smb_tran.o smb_trantcp.o smb_usr.o \
1297      subr_mchain.o

1299  SMBFS_COMMON_OBJS += smbfs_ntacl.o
1300  SMBFS_OBJS += smbfs_vfsops.o smbfs_vnops.o smbfs_node.o \
1301      smbfs_acl.o smbfs_client.o smbfs_smb.o \
1302      smbfs_subr.o smbfs_subr2.o \
1303      smbfs_rwlock.o smbfs_xattr.o \
1304      $(SMBFS_COMMON_OBJS)

1307 #
1308 #
1309 #
1310  MD_OBJS += md.o md_error.o md_ioctl.o md_mddb.o md_names.o \
1311      md_med.o md_rename.o md_subr.o

1313  MD_COMMON_OBJS = md_convert.o md_crc.o md_revchk.o

1315  MD_DERIVED_OBJS = metamed_xdr.o meta_basic_xdr.o

```

```

1317 SOFTPART_OBJS += sp.o sp_ioctl.o
1319 STRIPE_OBJS += stripe.o stripe_ioctl.o
1321 HOTSPARES_OBJS += hotspares.o
1323 RAID_OBJS += raid.o raid_ioctl.o raid_replay.o raid_resync.o raid_hotspare.o
1325 MIRROR_OBJS += mirror.o mirror_ioctl.o mirror_resync.o
1327 NOTIFY_OBJS += md_notify.o
1329 TRANS_OBJS += mdtrans.o trans_ioctl.o trans_log.o

1331 ZFS_COMMON_OBJS += \
1332     arc.o \
1333     bplist.o \
1334     bpobj.o \
1335     bptree.o \
1336     dbuf.o \
1337     ddt.o \
1338     ddt_zap.o \
1339     dmuf.o \
1340     dmuf_diff.o \
1341     dmuf_send.o \
1342     dmuf_object.o \
1343     dmuf_objset.o \
1344     dmuf_traverse.o \
1345     dmuf_tx.o \
1346     dnode.o \
1347     dnode_sync.o \
1348     dsl_dir.o \
1349     dsl_dataset.o \
1350     dsl_deadlist.o \
1351     dsl_destroy.o \
1352     dsl_pool.o \
1353     dsl_synctask.o \
1354     dsl_userhold.o \
1355     dmuf_zfetch.o \
1356     dsl_deleg.o \
1357     dsl_prop.o \
1358     dsl_scan.o \
1359     zfeature.o \
1360     gzip.o \
1361     lz4.o \
1362     lzjb.o \
1363     metaslab.o \
1364     refcount.o \
1365     rrwlock.o \
1366     sa.o \
1367     sha256.o \
1368     spa.o \
1369     spa_config.o \
1370     spa_errlog.o \
1371     spa_history.o \
1372     spa_misc.o \
1373     space_map.o \
1374     txg.o \
1375     uberblock.o \
1376     unique.o \
1377     vdev.o \
1378     vdev_cache.o \
1379     vdev_file.o \
1380     vdev_label.o \
1381     vdev_mirror.o \

```

```

1382     vdev_missing.o \
1383     vdev_queue.o \
1384     vdev_raidz.o \
1385     vdev_root.o \
1386     zap.o \
1387     zap_leaf.o \
1388     zap_micro.o \
1389     zfs_byteswap.o \
1390     zfs_debug.o \
1391     zfs_fm.o \
1392     zfs_fuid.o \
1393     zfs_sa.o \
1394     zfs_znode.o \
1395     zil.o \
1396     zio.o \
1397     zio_checksum.o \
1398     zio_compress.o \
1399     zio_inject.o \
1400     zle.o \
1401     zrlock.o

1403 ZFS_SHARED_OBJS += \
1404     zfeature_common.o \
1405     zfs_comutil.o \
1406     zfs_deleg.o \
1407     zfs_fletcher.o \
1408     zfs_namecheck.o \
1409     zfs_prop.o \
1410     zpool_prop.o \
1411     zprop_common.o

1413 ZFS_OBJS += \
1414     $(ZFS_COMMON_OBJS) \
1415     $(ZFS_SHARED_OBJS) \
1416     vdev_disk.o \
1417     zfs_acl.o \
1418     zfs_ctldir.o \
1419     zfs_dir.o \
1420     zfs_ioctl.o \
1421     zfs_log.o \
1422     zfs_onexit.o \
1423     zfs_replay.o \
1424     zfs_rlock.o \
1425     zfs_vfsops.o \
1426     zfs_vnops.o \
1427     zvol.o

1429 ZUT_OBJS += \
1430     zut.o

1432 #
1433 # streams modules
1434 #
1435 BUFMOD_OBJS += bufmod.o

1437 CONNLD_OBJS += connld.o

1439 DEDUMP_OBJS += dedump.o

1441 DRCOMPAT_OBJS += drcompat.o

1443 LDLINUX_OBJS += ldlinux.o

1445 LDTERM_OBJS += ldterm.o uwidth.o

1447 PCKT_OBJS += pckt.o

```

```

1449 PFMOD_OBJS +=    pfmod.o
1451 PTEM_OBJS +=     ptem.o
1453 REDIRMOD_OBJS +=  strredirm.o
1455 TIMOD_OBJS +=     timod.o
1457 TIRDWR_OBJS +=    tirdwr.o
1459 TTCOMPAT_OBJS +=  ttcompat.o
1461 LOG_OBJS +=       log.o
1463 PIPEMOD_OBJS +=    pipemod.o
1465 RPCMOD_OBJS +=     rpcmod.o      clnt_cots.o      clnt_clts.o \
1466                        clnt_gen.o      clnt_perr.o      mt_rpcinit.o      rpc_calmsg.o \
1467                        rpc_prot.o      rpc_sztypes.o   rpc_subr.o         rpcb_prot.o \
1468                        svc.o           svc_clts.o      svc_gen.o          svc_cots.o \
1469                        rpcsys.o        xdr_sizeof.o   clnt_rdma.o        svc_rdma.o \
1470                        xdr_rdma.o      rdma_subr.o     xdrdma_sizeof.o
1472 KLMMOD_OBJS +=      klmmod.o \
1473                        nlm_impl.o \
1474                        nlm_rpc_handle.o \
1475                        nlm_dispatch.o \
1476                        nlm_rpc_svc.o \
1477                        nlm_client.o \
1478                        nlm_service.o \
1479                        nlm_prot_clnt.o \
1480                        nlm_prot_xdr.o \
1481                        nlm_rpc_clnt.o \
1482                        nsm_addr_clnt.o \
1483                        nsm_addr_xdr.o \
1484                        sm_inter_clnt.o \
1485                        sm_inter_xdr.o
1487 KLMOPS_OBJS +=      klmops.o
1489 TLIMOD_OBJS +=      tlimod.o      t_kalloc.o      t_kbind.o      t_kclose.o \
1490                        t_kconnect.o  t_kfree.o       t_kgtstate.o   t_kopen.o \
1491                        t_krcvudat.o  t_ksndudat.o    t_kspoll.o     t_kunbind.o \
1492                        t_kutil.o
1494 RLMOD_OBJS +=        rlmmod.o
1496 TELMOD_OBJS +=       telmod.o
1498 CRYPTMOD_OBJS +=     cryptmod.o
1500 KB_OBJS +=           kbd.o          keytables.o
1502 #
1503 #                               ID mapping module
1504 #
1505 IDMAP_OBJS +=         idmap_mod.o    idmap_kapi.o    idmap_xdr.o     idmap_cache.o
1507 #
1508 #                               scheduling class modules
1509 #
1510 SDC_OBJS +=           sysdc.o
1512 RT_OBJS +=            rt.o
1513 RT_DPTBL_OBJS +=      rt_dptbl.o

```

```

1515 TS_OBJS +=           ts.o
1516 TS_DPTBL_OBJS +=     ts_dptbl.o
1518 IA_OBJS +=           ia.o
1520 FSS_OBJS +=          fss.o
1522 FX_OBJS +=           fx.o
1523 FX_DPTBL_OBJS +=     fx_dptbl.o
1525 #
1526 #                               Inter-Process Communication (IPC) modules
1527 #
1528 IPC_OBJS +=           ipc.o
1530 IPCMSG_OBJS +=        msg.o
1532 IPCSEM_OBJS +=        sem.o
1534 IPCSHM_OBJS +=        shm.o
1536 #
1537 #                               bignum module
1538 #
1539 COMMON_BIGNUM_OBJS += bignum_mod.o bignumimpl.o
1541 BIGNUM_OBJS += $(COMMON_BIGNUM_OBJS) $(BIGNUM_PSR_OBJS)
1543 #
1544 #                               kernel cryptographic framework
1545 #
1546 KCF_OBJS +=           kcf.o kcf_callprov.o kcf_cbufcall.o kcf_cipher.o kcf_crypto.o \
1547                        kcf_cryptoadm.o kcf_ctxops.o kcf_digest.o kcf_dual.o \
1548                        kcf_keys.o kcf_mac.o kcf_mech_tabs.o kcf_miscapi.o \
1549                        kcf_object.o kcf_policy.o kcf_prov_lib.o kcf_prov_tabs.o \
1550                        kcf_sched.o kcf_session.o kcf_sign.o kcf_spi.o kcf_verify.o \
1551                        kcf_random.o modes.o ecb.o cbc.o ctr.o ccm.o gcm.o \
1552                        fips_random.o
1554 CRYPTOADM_OBJS +=     cryptoadm.o
1556 CRYPTO_OBJS +=        crypto.o
1558 DPROV_OBJS +=         dprov.o
1560 DCA_OBJS +=           dca.o dca_3des.o dca_debug.o dca_dsa.o dca_kstat.o dca_rng.o \
1561                        dca_rsa.o
1563 AESPROV_OBJS +=       aes.o aes_impl.o aes_modes.o
1565 ARCFOURPROV_OBJS +=   arcfour.o arcfour_crypt.o
1567 BLOWFISHPROV_OBJS +=  blowfish.o blowfish_impl.o
1569 ECCPROV_OBJS +=       ecc.o ec.o ec2_163.o ec2_mont.o ecdecode.o ecl_mult.o \
1570                        ecp_384.o ecp_jac.o ec2_193.o ecl.o ecp_192.o ecp_521.o \
1571                        ecp_jm.o ec2_233.o ecl_curve.o ecp_224.o ecp_aff.o \
1572                        ecp_mont.o ec2_aff.o ec_naf.o ecl_gf.o ecp_256.o mp_gf2m.o \
1573                        mpi.o mplogic.o mpmontg.o mpprime.o oid.o \
1574                        secitem.o ec2_test.o ecp_test.o
1576 RSAPROV_OBJS +=       rsa.o rsa_impl.o pkcs1.o
1578 SWRANDPROV_OBJS +=    swrand.o

```

```

1580 #
1581 #             kernel SSL
1582 #
1583 KSSL_OBJS +=      kssl.o ksslioctl.o

1585 KSSL_SOCKETFIL_MOD_OBJS += ksslfilter.o ksslapi.o ksslrec.o

1587 #
1588 #             misc. modules
1589 #

1591 C2AUDIT_OBJS +=  adr.o audit.o audit_event.o audit_io.o \
1592                  audit_path.o audit_start.o audit_syscalls.o audit_token.o \
1593                  audit_mem.o

1595 PCIC_OBJS +=      pcic.o

1597 RPCSEC_OBJS +=    secmod.o          sec_clnt.o          sec_svc.o          sec_gen.o \
1598                  auth_des.o          auth_kern.o          auth_none.o          auth_loopb.o \
1599                  authdesprt.o        authdesubr.o        authu_prot.o \
1600                  key_call.o          key_prot.o          svc_authu.o          svcauthdes.o

1602 RPCSEC_GSS_OBJS +=      rpcsec_gssmod.o rpcsec_gss.o rpcsec_gss_misc.o \
1603                  rpcsec_gss_utils.o svc_rpcsec_gss.o

1605 CONSCONFIG_OBJS +=      consconfig.o

1607 CONSCONFIG_DACF_OBJS += consconfig_dacf.o consplat.o

1609 TEM_OBJS +=      tem.o tem_safe.o 6x10.o 7x14.o 12x22.o

1611 KBTRANS_OBJS +=      \
1612                  kbtrans.o \
1613                  kbtrans_keytables.o \
1614                  kbtrans_polled.o \
1615                  kbtrans_streams.o \
1616                  usb_keytables.o

1618 KGSSD_OBJS +=      gssd_clnt_stubs.o gssd_handle.o gssd_prot.o \
1619                  gss_display_name.o gss_release_name.o gss_import_name.o \
1620                  gss_release_buffer.o gss_release_oid_set.o gen_oids.o gssdmod.o

1622 KGSSD_DERIVED_OBJS = gssd_xdr.o

1624 KGSS_DUMMY_OBJS +=  dmech.o

1626 KSOCKET_OBJS +=      ksocket.o ksocket_mod.o

1628 CRYPTO= cksumtypes.o decrypt.o encrypt.o encrypt_length.o etypes.o \
1629          nfold.o verify_checksum.o prng.o block_size.o make_checksum.o \
1630          checksum_length.o hmac.o default_state.o mandatory_sumtype.o

1632 # crypto/des
1633 CRYPTO_DES= f_cbc.o f_cksum.o f_parity.o weak_key.o d3_cbc.o ef_crypto.o

1635 CRYPTO_DK= checksum.o derive.o dk_decrypt.o dk_encrypt.o

1637 CRYPTO_ARCFOUR= k5_arcfour.o

1639 # crypto/enc_provider
1640 CRYPTO_ENC= des.o des3.o arcfour_provider.o aes_provider.o

1642 # crypto/hash_provider
1643 CRYPTO_HASH= hash_kef_generic.o hash_kmd5.o hash_crc32.o hash_ksha1.o

1645 # crypto/keyhash_provider

```

```

1646 CRYPTO_KEYHASH= descbc.o k5_kmd5des.o k_hmac_md5.o

1648 # crypto/crc32
1649 CRYPTO_CRC32= crc32.o

1651 # crypto/old
1652 CRYPTO_OLD= old_decrypt.o old_encrypt.o

1654 # crypto/raw
1655 CRYPTO_RAW= raw_decrypt.o raw_encrypt.o

1657 K5_KRB= kfree.o copy_key.o \
1658          parse.o init_ctx.o \
1659          ser_adata.o ser_addr.o \
1660          ser_auth.o ser_cksum.o \
1661          ser_key.o ser_princ.o \
1662          serialize.o unparse.o \
1663          ser_actx.o

1665 K5_OS= timeofday.o toffset.o \
1666         init_os_ctx.o c_ustime.o

1668 SEAL= seal.o unseal.o

1670 MECH= delete_sec_context.o \
1671        import_sec_context.o \
1672        gssapi_krb5.o \
1673        k5seal.o k5unseal.o k5sealv3.o \
1674        ser_sctx.o \
1675        sign.o \
1676        util_crypt.o \
1677        util_validate.o util_ordering.o \
1678        util_seqnum.o util_set.o util_seed.o \
1679        wrap_size_limit.o verify.o

1683 MECH_GEN= util_token.o

1686 KGSS_KRB5_OBJS += krb5mech.o \
1687                  $(MECH) $(SEAL) $(MECH_GEN) \
1688                  $(CRYPTO) $(CRYPTO_DES) $(CRYPTO_DK) $(CRYPTO_ARCFOUR) \
1689                  $(CRYPTO_ENC) $(CRYPTO_HASH) \
1690                  $(CRYPTO_KEYHASH) $(CRYPTO_CRC32) \
1691                  $(CRYPTO_OLD) \
1692                  $(CRYPTO_RAW) $(K5_KRB) $(K5_OS)

1694 DES_OBJS +=      des_crypt.o des_impl.o des_ks.o des_soft.o

1696 DLBOOT_OBJS +=   bootparam_xdr.o nfs_dlinet.o scan.o

1698 KRTLD_OBJS +=     kobj_bootflags.o getoptstr.o \
1699                  kobj.o kobj_kdi.o kobj_lm.o kobj_subr.o

1701 MOD_OBJS +=       modctl.o modsubr.o modsysfile.o modconf.o modhash.o

1703 STRPLUMB_OBJS +=  strplumb.o

1705 CPR_OBJS +=        cpr_driver.o cpr_dump.o \
1706                  cpr_main.o cpr_misc.o cpr_mod.o cpr_stat.o \
1707                  cpr_uthread.o

1709 PROF_OBJS +=       prf.o

1711 SE_OBJS +=          se_driver.o

```



```

1713 SYSACCT_OBJS += acct.o
1715 ACCTCTL_OBJS += acctctl.o
1717 EXACCTSYS_OBJS += exacctsyst.o
1719 KATIO_OBJS += aio.o
1721 PCMCIA_OBJS += pcmcia.o cs.o cis.o cis_callout.o cis_handlers.o cis_params.o
1723 BUSRA_OBJS += busra.o
1725 PCS_OBJS += pcs.o
1727 PCAN_OBJS += pcana.o
1729 PCATA_OBJS += pcide.o pcdisk.o pclabel.o pcata.o
1731 PCSER_OBJS += pcser.o pcser_cis.o
1733 PCWL_OBJS += pcwl.o
1735 PSET_OBJS += pset.o
1737 OHCI_OBJS += ohci.o ohci_hub.o ohci_polled.o
1739 UHCI_OBJS += uhci.o uhciutil.o uhcitgt.o uhcihub.o uhcipolled.o
1741 EHCI_OBJS += ehci.o ehci_hub.o ehci_xfer.o ehci_intr.o ehci_util.o ehci_polled.o
1743 HUBD_OBJS += hubd.o
1745 USB_MID_OBJS += usb_mid.o
1747 USB_IA_OBJS += usb_ia.o
1749 UWBA_OBJS += uwba.o uwba_i.o
1751 SCESA2USB_OBJS += scsa2usb.o usb_ms_bulkonly.o usb_ms_cbi.o
1753 HWAHC_OBJS += hwahc.o hwahc_util.o
1755 WUSB_DF_OBJS += wusb_df.o
1756 WUSB_FWMOD_OBJS += wusb_fwmod.o
1758 IPF_OBJS += ip_fil_solaris.o fil.o solaris.o ip_state.o ip_frag.o ip_nat.o \
1759 ip_proxy.o ip_auth.o ip_pool.o ip_htable.o ip_lookup.o \
1760 ip_log.o misc.o ip_compat.o ip_nat6.o drand48.o
1762 IBD_OBJS += ibd.o ibd_cm.o
1764 EIBNX_OBJS += enx_main.o enx_hdlrs.o enx_ibt.o enx_log.o enx_fip.o \
1765 enx_misc.o enx_q.o enx_ctl.o
1767 EOIB_OBJS += eib_adm.o eib_chan.o eib_cm.o eib_ctl.o eib_data.o \
1768 eib_fip.o eib_ibt.o eib_log.o eib_mac.o eib_main.o \
1769 eib_rsrc.o eib_svc.o eib_vnic.o
1771 DLPSTUB_OBJS += dlpstub.o
1773 SDP_OBJS += sdpddi.o
1775 TRILL_OBJS += trill.o
1777 CTF_OBJS += ctf_create.o ctf_decl.o ctf_error.o ctf_hash.o ctf_labels.o \

```

```

1778 ctf_lookup.o ctf_open.o ctf_types.o ctf_util.o ctf_subr.o ctf_mod.o
1780 SMBIOS_OBJS += smb_error.o smb_info.o smb_open.o smb_subr.o smb_dev.o
1782 RPCIB_OBJS += rpcib.o
1784 KMDB_OBJS += kdrv.o
1786 AFE_OBJS += afe.o
1788 BGE_OBJS += bge_main2.o bge_chip2.o bge_kstats.o bge_log.o bge_ndd.o \
1789 bge_atomic.o bge_mii.o bge_send.o bge_recv2.o bge_mii_5906.o
1791 DMFE_OBJS += dmfe_log.o dmfe_main.o dmfe_mii.o
1793 EFE_OBJS += efe.o
1795 ELXL_OBJS += elxl.o
1797 HME_OBJS += hme.o
1799 IXGB_OBJS += ixgb.o ixgb_atomic.o ixgb_chip.o ixgb_gld.o ixgb_kstats.o \
1800 ixgb_log.o ixgb_ndd.o ixgb_rx.o ixgb_tx.o ixgb_xmii.o
1802 NGE_OBJS += nge_main.o nge_atomic.o nge_chip.o nge_ndd.o nge_kstats.o \
1803 nge_log.o nge_rx.o nge_tx.o nge_xmii.o
1805 PCN_OBJS += pcn.o
1807 RGE_OBJS += rge_main.o rge_chip.o rge_ndd.o rge_kstats.o rge_log.o rge_rxtx.o
1809 URTW_OBJS += urtw.o
1811 ARN_OBJS += arn_hw.o arn_eeprom.o arn_mac.o arn_calib.o arn_ani.o arn_phy.o arn_
1812 arn_main.o arn_recv.o arn_xmit.o arn_rc.o
1814 ATH_OBJS += ath_aux.o ath_main.o ath_osdep.o ath_rate.o
1816 ATU_OBJS += atu.o
1818 IPW_OBJS += ipw2100_hw.o ipw2100.o
1820 IWI_OBJS += ipw2200_hw.o ipw2200.o
1822 IWH_OBJS += iwh.o
1824 IWK_OBJS += iwk2.o
1826 IWP_OBJS += iwp.o
1828 MWL_OBJS += mwl.o
1830 MWLFW_OBJS += mwlfw_mode.o
1832 WPI_OBJS += wpi.o
1834 RAL_OBJS += rt2560.o ral_rate.o
1836 RUM_OBJS += rum.o
1838 RWD_OBJS += rt2661.o
1840 RWN_OBJS += rt2860.o
1842 UATH_OBJS += uath.o

```

```

1844 UATHFW_OBJS += uathfw_mod.o
1846 URAL_OBJS += ural.o
1848 RTW_OBJS += rtw.o smc93cx6.o rtwphy.o rtwphyio.o
1850 ZYD_OBJS += zyd.o zyd_usb.o zyd_hw.o zyd_fw.o
1852 MXFE_OBJS += mxfe.o
1854 MPTSAS_OBJS += mptsas.o mptsas_impl.o mptsas_init.o mptsas_raid.o mptsas_smhba.o
1856 SFE_OBJS += sfe.o sfe_util.o
1858 BFE_OBJS += bfe.o
1860 BRIDGE_OBJS += bridge.o
1862 IDM_SHARED_OBJS += base64.o
1864 IDM_OBJS += $(IDM_SHARED_OBJS) \
1865             idm.o idm_impl.o idm_text.o idm_conn_sm.o idm_so.o
1867 VR_OBJS += vr.o
1869 ATGE_OBJS += atge_main.o atge_l1e.o atge_mii.o atge_l1.o atge_l1c.o
1871 YGE_OBJS = yge.o
1873 #
1874 #       Build up defines and paths.
1875 #
1876 LINT_DEFS      += -Dunix
1878 #
1879 #       This duality can be removed when the native and target compilers
1880 #       are the same (or at least recognize the same command line syntax!)
1881 #       It is a bug in the current compilation system that the assembler
1882 #       can't process the -Y I, flag.
1883 #
1884 NATIVE_INC_PATH += $(INC_PATH) $(CCYFLAG)$ (UTSBASE)/common
1885 AS_INC_PATH     += $(INC_PATH) -I$(UTSBASE)/common
1886 INCLUDE_PATH    += $(INC_PATH) $(CCYFLAG)$ (UTSBASE)/common
1888 PCIEB_OBJS += pcieb.o
1890 #       Chelsio N110 10G NIC driver module
1891 #
1892 CH_OBJS = ch.o glue.o pe.o sge.o
1894 CH_COM_OBJS = ch_mac.o ch_subr.o csapi.o espi.o ix1010.o mc3.o mc4.o mc5.o \
1895             mv88e1xxx.o mv88x201x.o my3126.o pm3393.o tp.o ulp.o \
1896             vsc7321.o vsc7326.o xpak.o
1898 #
1899 #       Chelsio Terminator 4 10G NIC nexus driver module
1900 #
1901 CXGBE_FW_OBJS = t4_fw.o t4_cfg.o
1902 CXGBE_COM_OBJS = t4_hw.o common.o
1903 CXGBE_NEX_OBJS = t4_nexus.o t4_sge.o t4_mac.o t4_ioctl.o shared.o \
1904             t4_l2t.o adapter.o osdep.o
1906 #
1907 #       Chelsio Terminator 4 10G NIC driver module
1908 #
1909 CXGBE_OBJS = cxgbe.o

```

```

1911 #
1912 #       PCI strings file
1913 #
1914 PCI_STRING_OBJS = pci_strings.o
1916 NET_DACF_OBJS += net_dacf.o
1918 #
1919 #       Xframe 10G NIC driver module
1920 #
1921 XGE_OBJS = xge.o xgell.o
1923 XGE_HAL_OBJS = xgehal-channel.o xgehal-fifo.o xgehal-ring.o xgehal-config.o \
1924             xgehal-driver.o xgehal-mm.o xgehal-stats.o xgehal-device.o \
1925             xge-queue.o xgehal-mgmt.o xgehal-mgmtaux.o
1927 #
1928 #       e1000g module
1929 #
1930 E1000G_OBJS += e1000_80003es2lan.o e1000_82540.o e1000_82541.o e1000_82542.o \
1931             e1000_82543.o e1000_82571.o e1000_api.o e1000_ich8lan.o \
1932             e1000_mac.o e1000_manage.o e1000_nvm.o e1000_osdep.o \
1933             e1000_phy.o e1000g_debug.o e1000g_main.o e1000g_alloc.o \
1934             e1000g_tx.o e1000g_rx.o e1000g_stat.o
1936 #
1937 #       Intel 82575 1G NIC driver module
1938 #
1939 IGB_OBJS = igb_82575.o igb_api.o igb_mac.o igb_manage.o \
1940             igb_nvm.o igb_osdep.o igb_phy.o igb_buf.o \
1941             igb_debug.o igb_gld.o igb_log.o igb_main.o \
1942             igb_rx.o igb_stat.o igb_tx.o
1944 #
1945 #       Intel Pro/100 NIC driver module
1946 #
1947 IPRB_OBJS = iprb.o
1949 #
1950 #       Intel 10GbE PCIE NIC driver module
1951 #
1952 IXGBE_OBJS = ixgbe_82598.o ixgbe_82599.o ixgbe_api.o \
1953             ixgbe_common.o ixgbe_phy.o \
1954             ixgbe_buf.o ixgbe_debug.o ixgbe_gld.o \
1955             ixgbe_log.o ixgbe_main.o \
1956             ixgbe_osdep.o ixgbe_rx.o ixgbe_stat.o \
1957             ixgbe_tx.o ixgbe_x540.o ixgbe_mbx.o
1959 #
1960 #       NIU 10G/16 driver module
1961 #
1962 NXGE_OBJS = nxge_mac.o nxge_ipp.o nxge_rxdma.o \
1963             nxge_txdma.o nxge_txc.o nxge_main.o \
1964             nxge_hw.o nxge_fzc.o nxge_virtual.o \
1965             nxge_send.o nxge_classify.o nxge_fflp.o \
1966             nxge_fflp_hash.o nxge_ndd.o nxge_kstats.o \
1967             nxge_zcp.o nxge_fm.o nxge_espc.o nxge_hv.o \
1968             nxge_hio.o nxge_hio_guest.o nxge_intr.o
1970 NXGE_NPI_OBJS = \
1971             npi.o npi_mac.o npi_ipp.o \
1972             npi_txdma.o npi_rxdma.o npi_txc.o \
1973             npi_zcp.o npi_espc.o npi_fflp.o \
1974             npi_vir.o

```

```

1976 NXGE_HCALL_OBJS =      \
1977      nxge_hcall.o

1979 #
1980 # Virtio modules
1981 #

1983 # Virtio core
1984 VIRTIO_OBJS = virtio.o

1986 # Virtio block driver
1987 VIOBLK_OBJS = vioblk.o

1989 #
1990 #      kiconv modules
1991 #
1992 KICONV_EMEA_OBJS += kiconv_emea.o

1994 KICONV_JA_OBJS += kiconv_ja.o

1996 KICONV_KO_OBJS += kiconv_cck_common.o kiconv_ko.o

1998 KICONV_SC_OBJS += kiconv_cck_common.o kiconv_sc.o

2000 KICONV_TC_OBJS += kiconv_cck_common.o kiconv_tc.o

2002 #
2003 #      AAC module
2004 #
2005 AAC_OBJS = aac.o aac_ioctl.o

2007 #
2008 #      sdcard modules
2009 #
2010 SDA_OBJS =      sda_cmd.o sda_host.o sda_init.o sda_mem.o sda_mod.o sda_slot.o
2011 SDHOST_OBJS =   sdhost.o

2013 #
2014 #      hxge 10G driver module
2015 #
2016 HXGE_OBJS =      hxge_main.o hxge_vmac.o hxge_send.o      \
2017      hxge_txdma.o hxge_rxdma.o hxge_virtual.o      \
2018      hxge_fm.o hxge_fzc.o hxge_hw.o hxge_kstats.o      \
2019      hxge_ndd.o hxge_pfc.o      \
2020      hpi.o hpi_vmac.o hpi_rxdma.o hpi_txdma.o      \
2021      hpi_vir.o hpi_pfc.o

2023 #
2024 #      MEGARAID_SAS module
2025 #
2026 MEGA_SAS_OBJS = megaraid_sas.o

2028 #
2029 #      MR_SAS module
2030 #
2031 MR_SAS_OBJS = ld_pd_map.o mr_sas.o mr_sas_tbolt.o mr_sas_list.o

2033 #
2034 #      ISCSI_INITIATOR module
2035 #
2036 ISCSI_INITIATOR_OBJS = chap.o iscsi_io.o iscsi_thread.o      \
2037      iscsi_ioctl.o iscsid.o iscsi.o      \
2038      iscsi_login.o isns_client.o iscsiAuthClient.o      \
2039      iscsi_lun.o iscsiAuthClientGlue.o      \
2040      iscsi_net.o nvfile.o iscsi_cmd.o      \
2041      iscsi_queue.o persistent.o iscsi_conn.o \

```

```

2042      iscsi_sess.o radius_auth.o iscsi_crc.o \
2043      iscsi_stats.o radius_packet.o iscsi_doorctl.o \
2044      iscsi_targetparam.o utils.o kifconf.o

2046 #
2047 #      ntxn 10Gb/16b NIC driver module
2048 #
2049 NTXN_OBJS =      unm_nic_init.o unm_gem.o unm_nic_hw.o unm_ndd.o \
2050      unm_nic_main.o unm_nic_isr.o unm_nic_ctx.o niu.o

2052 #
2053 #      Myricom 10Gb NIC driver module
2054 #
2055 MYRI10GE_OBJS = myri10ge.o myri10ge_lro.o

2057 #      nulldriver module
2058 #
2059 NULLDRIVER_OBJS =      nulldriver.o

2061 TPM_OBJS =      tpm.o tpm_hcall.o

```

```
*****
```

```
28022 Sat Sep 14 15:00:33 2013
```

```
new/usr/src/uts/common/os/interrupt_balancer.c
```

```
XXX Rearchitect and replace interrupt distribution
```

```
*****
```

```
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 */

12 /*
13  * Copyright 2013 David Hoeppner. All rights reserved.
14 */

16 /*
17  * Interrupt Load Balancer.
18  *
19  * The original balance functions views all CPU as equally.
20 */

22 /* XXX
23  *
24  * ib_cpu_list::walk list |::print ib_cpu_t
25  *
26  *
27  */
28 */

29 #include <sys/param.h>
30 #include <sys/types.h>
31 #include <sys/systm.h>
32 #include <sys/callb.h>
33 #include <sys/cpuvar.h>
34 #include <sys/proc.h>
35 #include <sys/processor.h>
36 #include <sys/sdt.h>
37 #include <sys/sysmacros.h>
38 #include <sys/time.h>
39 #include <sys/cmn_err.h>
40 #include <sys/zone.h>
41 #include <sys/lgrp.h>

43 #include <sys/pci_tools.h>

45 extern proc_t *proc_intrd;

47 #define IB_NAME "intrd"

49 #define IS_CPU(cpu_id) (cpu[cpu_id] != NULL)

51 #define IB_NORMAL_SLEEPTIME 10
52 #define IB_IDLE_SLEEPTIME 45
53 #define IB_ONECPU_SLEEPTIME (60 * 15)

55 #define IB_NUM_SAMPLES 6

58 static kmutex_t ib_lock;
59 static kcondvar_t ib_cv;

61 */
```

```
62 * Interrupt CPU instance.
63 */
64 typedef struct _ib_cpu {
65     list_node_t ic_next;
66     boolean_t ic_offline;

68     hrtime_t ic_tot;
69     list_t ic_ivec_list;
70     uint32_t ic_num_ivecs;
71     processorid_t ic_cpu_id; /* XXX duplicate */
72     int64_t ic_intrs;
73     int64_t ic_big_intrs;
74     int64_t ic_bigintr; /* XXX bitintrs */

76     int ic_intr_load; /* intrs / tot */
77 } ib_cpu_t;

79 /*
80 * Interrupt vector instance.
81 */
82 typedef struct _ib_ivec {
83     list_node_t ii_next; /* link */

85     uint64_t ii_ihs;
86     uint64_t ii_ino;
87     uint64_t ii_num_ino;
88     uint64_t ii_pil;
89     uint64_t ii_time;
90     char *ii_buspath;
91     char *ii_name;

93     processorid_t ii_orig_cpu;
94     processorid_t ii_now_cpu;
95     uint64_t ii_inum;
96 } ib_ivec_t;

98 /*
99 * MSI
100 */
101 typedef struct _ib_msi {
102     list_node_t im_next; /* link */
103     const char *im_name;
104     list_t im_ino_list;
105 } ib_msi_t;

107 typedef struct _ib_msi_ino {
108     list_node_t imi_next; /* link */
109     uint64_t imi_ino;
110     ib_ivec_t *imi_ivec;
111 } ib_msi_ino_t;

113 /*
114 * Snapshot
115 */
116 typedef struct _ib_snapshot {
117     list_node_t is_next; /* link */
118     list_t is_cpu_list;
119     processorid_t is_num_cpus;
120 } ib_snapshot_t;

122 /*
123 * Snapshot delta structure.
124 */
125 typedef struct _ib_delta {
126     list_node_t id_next; /* link */
127     list_t id_cpu_list;
```

```

128     boolean_t    id_missing;
129     int          id_avgintrload; /* interrupts / total time */
130     uint64_t     id_avgintrnsec;
131     int          id_goodness;
132 } ib_delta_t;

135 static list_t   ib_cpu_list;          /* List of all CPU's */

137 static uint8_t  ib_cs = 0;            /* Index of current sample */
138 static long     ib_sleeptime = IB_NORMAL_SLEEPTIME;
139 static processorid_t  ib_num_cpus;

141 static int      goodness_unsafe_load = 90;
142 static int      goodness_mindelta = 10;

144 /*
145  * Function prototypes.
146  */
147 static void      ib_cpu_register(processorid_t);
148 static int       ib_cpu_setup(cpu_setup_t, int, void *);
149 static int       ib_goodness(ib_delta_t *);
150 static int       ib_do_reconfig(ib_delta_t *);
151 static int       ib_imbalanced(int, int);
152 static int       ib_interrupt_do_move(ib_ivec_t *, processorid_t);

154 static ib_snapshot_t *ib_get_statistics(void);
155 static ib_delta_t   *ib_delta_generate(ib_snapshot_t *, ib_snapshot_t *);

157 /*
158  * Helper macros.
159  */
160 #define FOREACH_CPU(icpu, icpu_list)          \
161     for (icpu = list_head(&icpu_list); icpu != NULL; \
162          icpu = list_next(&icpu_list, icpu))

164 #define FOREACH_IVEC(ivec, ivec_list)          \
165     for (ivec = list_head(&ivec_list); ivec != NULL; \
166          ivec = list_next(&ivec_list, ivec))

168 #define DTRACE_INTRD(name)          \
169     DTRACE_PROBE(__intrd_##name)

171 #define DEBUG 1
172 #ifdef DEBUG
173 #define IB_APIDBG(args) cmn_err args
174 #define IB_IMPLDBG(args) cmn_err args
175 #else
176 #define IB_APIDBG(args)
177 #define IB_IMPLDBG(args)
178 #endif

180 #define IB_LOG(args)      cmn_err args

182 void
183 interrupt_balancer(void)
184 {
185     processorid_t  cpu_id;
186     callb_cpr_t    cpr;
187     user_t          *u = PTOU(curproc);
188     int             error;

190     boolean_t      do_reconfig = B_FALSE;
191     int             goodness;
192     int             baseline_goodness = 0;
193     list_t          ib_delta_list;

```

```

194     hrtime_t       statslen = 60;

196     proc_intrd = ttoproc(curthread);
197     proc_intrd->p_cstime = proc_intrd->p_stime = 0;
198     proc_intrd->p_cutime = proc_intrd->p_utime = 0;

200     (void) strncpy(u->u_psargs, IB_NAME, sizeof(u->u_psargs));
201     (void) strncpy(u->u_comm, IB_NAME, sizeof(u->u_comm));

203     /* Initialize global mutex lock */
204     mutex_init(&ib_lock, NULL, MUTEX_DEFAULT, NULL);

206     /* Initialize CPU list */
207     list_create(&ib_cpu_list, sizeof (ib_cpu_t),
208                offsetof(ib_cpu_t, ic_next));

210     /* Initialize delta list */
211     list_create(&ib_delta_list, sizeof (ib_delta_t),
212                offsetof(ib_delta_t, id_next));

214     /*
215      * Build a list of all CPUs available for interrupt handling.
216      */
217     for (cpu_id = 0; cpu_id <= max_cpu_seqid_ever; cpu_id++) {
218         if (IS_CPU(cpu_id))
219             ib_cpu_register(cpu_id);
220     }

222     /*
223      * Locality group information.
224      */
225     int i;
226     for (i = 0; i < lgrp_plat_max_lgrps(); i++) {
227         lgrp_t *lgrp;

229         lgrp = lgrp_table[i];
230     }

232     /*
233      * Register a callback if a CPU goes offline or comes online.
234      */
235     mutex_enter(&cpu_lock);
236     register_cpu_setup_func(ib_cpu_setup, NULL);
237     mutex_exit(&cpu_lock);

239     CALLB_CPR_INIT(&cpr, &ib_lock, callb_generic_cpr, IB_NAME);

241     ib_snapshot_t *snapshot = NULL;
242     ib_snapshot_t *new_snapshot = NULL;
243     hrtime_t      delta_time;
244     hrtime_t      deltas_tottime = 0;
245     boolean_t     below_statslen;

247     snapshot = ib_get_statistics();

249     mutex_enter(&ib_lock);
250     for (;;) {
251         ib_delta_t *delta;

253         DTRACE_INTRD(get_stats);
254         new_snapshot = ib_get_statistics();

256         delta = ib_delta_generate(snapshot, new_snapshot);

258         below_statslen = (deltas_tottime < statslen);
259         deltas_tottime += delta_time;

```

```

260     do_reconfig = (below_statslen && deltas_tottime >= statslen);
262     list_insert_tail(&ib_delta_list, delta);

264     /*
265      * Calculate the goodness of the current configuration.
266      */
267     goodness = ib_goodness(delta);

269     if (ib_imbalanced(goodness, baseline_goodness))
270         do_reconfig = B_TRUE;

272     /*
273      * Reconfigure interrupt distribution.
274      */
275     if (do_reconfig) {
276         error = ib_do_reconfig(delta);

278         if (error != 0) {
279             if (error == -1)
280                 IB_LOG((CE_CONT, "ib_do_reconfig failed!
281             } else {
282                 IB_LOG((CE_CONT, "setting new baseline of %d", g
283                 baseline_goodness = goodness;
284             }
285         }

287         /*
288          * Wait for timeout or CPU reconfiguration.
289          */
290         CALLB_CPR_SAFE_BEGIN(&cpr);
291         cv_timedwait(&ib_cv, &ib_lock, ddi_get_lbolt() +
292             SEC_TO_TICK(ib_sleeptime));
293         CALLB_CPR_SAFE_END(&cpr, &ib_lock);
294     }

296     CALLB_CPR_EXIT(&cpr);

298     /*
299      * Unregister CPU callback.
300      */
301     mutex_enter(&cpu_lock);
302     unregister_cpu_setup_func(ib_cpu_setup, NULL);
303     mutex_exit(&cpu_lock);
304 }

306 /*
307  * Register a new CPU in the global list of CPUs.
308  */
309 static void
310 ib_cpu_register(processorid_t cpu_id)
311 {
312     cpu_t          *cp = cpu[cpu_id];
313     ib_cpu_t       *new_cpu;

315     new_cpu = kmem_alloc(sizeof(ib_cpu_t), KM_SLEEP);
316     new_cpu->ic_cpu_id = cpu_id;

318     /* Initialize list for interrupt vectors */
319     list_create(&new_cpu->ic_ivec_list, sizeof(ib_ivec_t),
320         offsetof(ib_ivec_t, ii_next));

322     list_link_init(&new_cpu->ic_next);

324     /* Check if this CPU can handle interrupts */
325     mutex_enter(&cpu_lock);

```

```

326     if (cpu_is_nointr(cp))
327         new_cpu->ic_offline = B_TRUE;
328     else
329         new_cpu->ic_offline = B_FALSE;
330     mutex_exit(&cpu_lock);

332     /* Add CPU to list of CPUs */
333     list_insert_tail(&ib_cpu_list, new_cpu);

335     ib_num_cpus++;

337     IB_IMPLDBG((CE_CONT, "ib_cpu_register: cpu=0x%x", cpu_id));
338 }

340 /*
341  * Unregister CPU from the global list of CPUs.
342  */
343 static void
344 ib_cpu_unregister(processorid_t cpu_id)
345 {
346     ib_cpu_t       *icpu;

348     mutex_enter(&ib_lock);
349     FOREACH_CPU(icpu, ib_cpu_list) {
350         if (icpu->ic_cpu_id == cpu_id) {
351             list_remove(&ib_cpu_list, icpu);
352             /* XXX or just offline CPU; statistics? */
353             break;
354         }
355     }
356     mutex_exit(&ib_lock);

358     ib_num_cpus--;

360     IB_IMPLDBG((CE_CONT, "ib_cpu_unregister: cpu=0x%x",
361         cpu_id));
362 }

364 /*
365  * Hook for CPU changes.
366  */
367 static int
368 ib_cpu_setup(cpu_setup_t what, int cpu_id, void *arg)
369 {
371     switch (what) {
372         /* XXX */
373         case CPU_OFF:
374             ib_cpu_unregister(cpu_id);
375             cv_signal(&ib_cv);
376             break;

378         case CPU_INTR_ON:
379             ib_cpu_register(cpu_id);
380             cv_signal(&ib_cv);
381             break;

383         default:
384             break;
385     }

387     return (0);
388 }

390 static ib_cpu_t *
391 ib_cpu_create(void)

```

```

392 {
393     ib_cpu_t      *icpu;

395     icpu = kmem_alloc(sizeof (ib_cpu_t), KM_SLEEP);

397     return (icpu);
398 }

400 /*
401  * Find a CPU in the global list of CPUs by processor id.
402  */
403 static ib_cpu_t *
404 ib_cpu_find(list_t cpu_list, processorid_t cpu_id)
405 {
406     ib_cpu_t      *icpu;

408     IB_APIDBG((CE_CONT, "ib_cpu_find: API cpu = %d", cpu_id));

410     FOREACH_CPU(icpu, cpu_list) {
411         if (icpu->ic_cpu_id == cpu_id)
412             return (icpu);
413     }

415     return (NULL);
416 }

418 /*
419  * Find a interrupt vector for a specific CPU.
420  */
421 static ib_ivec_t *
422 ib_cpu_find_ivec(list_t cpu_list, processorid_t cpu_id, char *buspath, uint64_t
423 {
424     ib_cpu_t      *icpu;
425     ib_ivec_t      *ivec;

427     icpu = ib_cpu_find(cpu_list, cpu_id);
428     if (icpu == NULL)
429         return (NULL);

431     for (ivec = list_head(&icpu->ic_ivec_list); ivec != NULL;
432          ivec = list_next(&icpu->ic_ivec_list, ivec)) {
433         if (ivec->ii_ino == ino)
434             return (ivec);
435     }

437     return (NULL);
438 }

440 /*
441  * Total times spend.
442  */
443 static void
444 ib_cpu_statistics(ib_cpu_t *icpu)
445 {
446     cpu_t          *cp;
447     hrtime_t        msnssecs[NCMSTATES];
448     hrtime_t        new_tot;

450     cp = cpu[icpu->ic_cpu_id];
451     get_cpu_mstate(cp, msnssecs);

453     icpu->ic_tot = msnssecs[CMS_IDLE] + msnssecs[CMS_USER] +
454                 msnssecs[CMS_SYSTEM];

456 }

```

```

458 /*
459  * Create a new interrupt vector.
460  */
461 static ib_ivec_t *
462 ib_ivec_create(const char *buspath, uint64_t ino)
463 {
464     ib_ivec_t      *ivec;

466     ivec = (ib_ivec_t *)kmem_alloc(sizeof (ib_ivec_t), KM_SLEEP);
467     list_link_init(&ivec->ii_next);

470     ivec->ii_buspath = (char *)buspath;      /* XXX: strdup */
471     ivec->ii_ino = ino;
472     ivec->ii_ihs = 1;

474     return (ivec);
475 }

477 static void
478 intrd_ivec_register(ib_cpu_t *icpu)
479 {
480 }

482 /*
483  * Find interrupt vector by ino.
484  */
485 static ib_ivec_t *
486 ib_ivec_find_ino(list_t ivec_list, uint64_t ino)
487 {
488     ib_ivec_t      *ivec;

490     FOREACH_IVEC(ivec, ivec_list) {
491         if (ivec->ii_inum == ino)
492             return (ivec);
493     }

495     return (NULL);
496 }

498 /*
499  * Delete a interrupt vector from a list.
500  */
501 static void
502 ib_ivec_delete_ino(list_t ivec_list, uint64_t ino)
503 {
504     ib_ivec_t      *ivec;

506     FOREACH_IVEC(ivec, ivec_list) {
507         if (ivec->ii_inum == ino) {
508             /* XXX: remove from list */
509             ;
510         }
511     }
512 }

514 /*
515  * Add a new interrupt vector to a list.
516  */
517 static void
518 ib_ivec_add_ino(list_t ivec_list, ib_ivec_t *ivec)
519 {
520     list_insert_tail(&ivec_list, ivec);
521 }

523 static ib_msi_t *

```

```

524 ib_msi_create(const char *name)
525 {
526     ib_msi_t      *msi;
527
528     msi = (ib_msi_t *)kmem_alloc(sizeof (ib_msi_t), KM_SLEEP);
529
530     msi->im_name = name;
531
532     list_link_init(&msi->im_next);
533     list_create(&msi->im_ino_list, sizeof (ib_msi_ino_t),
534         offsetof(ib_msi_ino_t, imi_next));
535
536     return (msi);
537 }
538
539 /*
540  * Allocate and initialize a new snapshot structure.
541  */
542 static ib_snapshot_t *
543 ib_snapshot_create(void)
544 {
545     ib_snapshot_t  *snapshot;
546
547     snapshot = kmem_alloc(sizeof (ib_snapshot_t), KM_SLEEP);
548
549     /* init link */
550
551     /* Initialize CPU list */
552     list_create(&snapshot->is_cpu_list, sizeof (ib_cpu_t),
553         offsetof(ib_cpu_t, ic_next));
554
555     snapshot->is_num_cpus = 0;
556
557     return (snapshot);
558 }
559
560 static ib_ivec_t *
561 ib_irq_fill_ivec(kstat_t *ksp)
562 {
563     kstat_named_t  *knp;
564     ib_ivec_t      *ivec;
565     char           *datap;
566     uint64_t       time;
567     int            i;
568
569     datap = ksp->ks_data;
570     knp = KSTAT_NAMED_PTR(ksp);
571     for (i = 0; i < ksp->ks_ndata; i++, knp++) {
572         IB_IMPLDBG((CE_CONT, "ib_irq_fill_ivec: %s",
573             knp->name));
574
575         if (strcmp(knp->name, "time") == 0) {
576             cmn_err(CE_CONT, "XXX ib time");
577             time = knp->value.ui64;
578         }
579
580         knp += sizeof (kstat_named_t);
581         datap += sizeof (kstat_named_t);
582     }
583
584     /* Allocate a new interrupt vector */
585     ivec = ib_ivec_create("", 0);
586     ivec->ii_time = time;
587
588     return (ivec);
589 }

```

```

591 /*
592  * XXX: icpu not needed, move out of loop
593  */
594 static void
595 ib_irq_statistics(ib_cpu_t *icpu)
596 {
597     kstat_t      *ksp;
598     int          instance = 1;
599
600     /*
601      * Read pci interrupts.
602      */
603     ksp = kstat_hold_byname("pci_intrs", instance, "pci", ALL_ZONES);
604     while (ksp != NULL) {
605         KSTAT_ENTER(ksp);
606
607         if (ksp->ks_data != NULL && ksp->ks_type == KSTAT_TYPE_NAMED) {
608             ib_cpu_t      *icpu;
609             ib_ivec_t      *ivec;
610             kstat_named_t  *knp;
611             kstat_named_t  *datap;
612             uint64_t       ino;
613             char           *buspath;
614             char           *namep;
615             processorid_t  cpu_id;
616             int            i;
617             boolean_t      is_enabled = B_TRUE;
618
619             (void) KSTAT_UPDATE(ksp, KSTAT_READ);
620
621             /*
622              * Find the CPU this interrupt vector is on and
623              * if the vector itself is enabled.
624              */
625             datap = ksp->ks_data;
626             namep = KSTAT_NAMED_PTR(ksp)->name;
627             for (i = 0; i < ksp->ks_ndata; i++) {
628                 if (strcmp(namep, "cpu") == 0) {
629                     cpu_id = datap->value.ui64;
630                 } else if (strcmp(namep, "type") == 0) {
631                     if (strcmp(datap->value.c, "disabled") =
632                         is_enabled = B_FALSE;
633                     break;
634                 }
635             }
636
637             namep += sizeof (kstat_named_t);
638             datap += sizeof (kstat_named_t);
639         }
640
641         /*
642          * Skip this interrupt vector if its disabled.
643          */
644         if (!is_enabled)
645             continue;
646
647         /*
648          * Check if CPU is online.
649          */
650         icpu = ib_cpu_find(ib_cpu_list, cpu_id);
651         if (icpu == NULL || icpu->ic_offline)
652             continue;
653
654         /*
655          * Fill information.

```



```

656         */
657         ivec = ib_irq_fill_ivec(ksp);
658         if (ivec == NULL)
659             continue;
661         list_insert_tail(&icpu->ic_ivec_list, ivec);
662     }
664     KSTAT_EXIT(ksp);
665     kstat_rele(ksp);
667     instance++;
668     ksp = kstat_hold_byname("pci_intrs", instance, "pci", ALL_ZONES)
669 }
670 }
672 /*
673  * Collect data from CPUs and interrupt vectors.
674  */
675 static ib_snapshot_t *
676 ib_get_statistics(void)
677 {
678     ib_cpu_t      *os_cpu;
679     ib_snapshot_t *snapshot;
680     ib_cpu_t      *snapshot_cpu;
682     /*
683      * Nothing to balance with one CPU. XXX: right place?
684      */
685     if (ib_num_cpus <= 1) {
686         ib_sleeptime = IB_ONECPU_SLEEPTIME;
687         return (NULL);
688     }
690     /*
691      * Store all CPUs and ivecs here.
692      */
693     snapshot = ib_snapshot_create();
695     /*
696      * Loop over all active CPUs
697      */
698     FOREACH_CPU(os_cpu, ib_cpu_list) {
700         snapshot->is_num_cpus++;
702         snapshot_cpu = ib_cpu_create();
703         snapshot_cpu->ic_cpu_id = os_cpu->ic_cpu_id;
705         list_insert_tail(&snapshot->is_cpu_list, snapshot_cpu);
707         ib_cpu_statistics(snapshot_cpu);
708         ib_irq_statistics(os_cpu);
709     }
711     return (snapshot);
712 }
714 static ib_delta_t *
715 ib_delta_create(void)
716 {
717     ib_delta_t      *delta;
719     delta = kmem_alloc(sizeof (ib_delta_t), KM_SLEEP);
720     delta->id_missing = B_FALSE;

```

```

722     list_create(&delta->id_cpu_list, sizeof (ib_cpu_t),
723         offsetof(ib_cpu_t, ic_next));
725     return (delta);
726 }
728 /*
729  * Generate the delta of two snapshots.
730  */
731 static ib_delta_t *
732 ib_delta_generate(ib_snapshot_t *old_snapshot, ib_snapshot_t *new_snapshot)
733 {
734     ib_cpu_t      *old_cpu, *new_cpu;
735     ib_delta_t      *delta;
736     int            intrload = 0;
737     int            intrnsec = 0;
738     processorid_t  cpus = 0;
740     /*
741      * Allocate a new delta structure.
742      */
743     delta = ib_delta_create();
745     /*
746      * Number of CPUs must be the same.
747      */
748     delta->id_missing = old_snapshot->is_num_cpus !=
749         new_snapshot->is_num_cpus;
751     if (delta->id_missing != 0) {
752         IB_LOG((CE_CONT, "ib_delta_generate: number of CPUs changed"));
753         return (delta);
754     }
756     /*
757      * Loop over the CPUs in both snapshots.
758      */
759     for (new_cpu = list_head(&new_snapshot->is_cpu_list),
760         old_cpu = list_head(&old_snapshot->is_cpu_list);
761         new_cpu != NULL && old_cpu != NULL;
762         new_cpu = list_next(&new_snapshot->is_cpu_list, new_cpu),
763         old_cpu = list_next(&old_snapshot->is_cpu_list, old_cpu)) {
764         ib_cpu_t      *delta_cpu;
765         ib_ivec_t      *new_ivec;
767         /* XXX: just onlined CPU? */
769         /* Allocate a new CPU structure */
770         delta_cpu = ib_cpu_create();
772         /* Difference of total time */
773         delta_cpu->ic_tot = new_cpu->ic_tot - old_cpu->ic_tot;
774         if (!(delta_cpu->ic_tot >= 0)) {
775             delta->id_missing = B_TRUE;
776             kmem_free(delta_cpu, sizeof (ib_cpu_t));
777             return (delta);
778         }
780         list_insert_tail(&delta->id_cpu_list, delta_cpu);
782         /* Avoid division by zero */
783         if (delta_cpu->ic_tot == 0)
784             delta_cpu->ic_tot = 1;
786         delta_cpu->ic_intrs = 0;
787         delta_cpu->ic_big_intrs = 0;

```

```

789      /*
790      * Number of interrupt vectors must be the same.
791      */
792      if (old_cpu->ic_num_ivecs != new_cpu->ic_num_ivecs) {
793          IB_LOG((CE_CONT, "ib_delta_generate: cpu %d has more "
794              "or less interrupts", old_cpu->ic_cpu_id));
795          delta->id_missing = B_TRUE;
796          return (delta);
797      }

799      /*
800      * Loop over the interrupt vectors of the new CPU.
801      */
802      for (new_ivec = list_head(&new_cpu->ic_ivec_list);
803          new_ivec != NULL; new_ivec =
804              list_next(&new_cpu->ic_ivec_list, new_ivec)) {
805          ib_ivec_t      *ivec;
806          ib_ivec_t      *delta_ivec;
807          hrtime_t        time;

809          if (new_ivec->ii_num_ino == 0)
810              continue;

812          /*
813          * If interrupt vector does not exists or XXX crtime
814          * is different, set missing.
815          */
816          ivec = ib_ivec_find_ino(old_cpu->ic_ivec_list,
817              new_ivec->ii_ino);
818          if (ivec == NULL) {
819              delta->id_missing = B_TRUE;
820              return (delta);
821          }

823          /* Allocate a new delta interrupt vector */
824          delta_ivec = ib_ivec_create(new_ivec->ii_buspath,
825              new_ivec->ii_ino);

827          /*
828          * Time used by this interrupt.
829          */
830          time = new_ivec->ii_time - ivec->ii_time;
831          if (time < 0) {
832              delta->id_missing = B_TRUE;
833              kmem_free(delta_ivec, sizeof (ib_delta_t));
834              return (delta);
835          }

837          delta_cpu->ic_intrs += time;
838          delta_ivec->ii_time = time;

840          if (time > delta_cpu->ic_bigintr)
841              delta_cpu->ic_bigintr = time;

843          /*
844          * Fill in the rest.
845          */
846          delta_ivec->ii_ihs = new_ivec->ii_ihs;
847          delta_ivec->ii_pil = new_ivec->ii_pil;
848          delta_ivec->ii_ino = new_ivec->ii_ino;
849          delta_ivec->ii_num_ino = new_ivec->ii_num_ino;
850          /* XXX: buspath, name */
851      }
853      /*

```

```

854          * Rounding error
855          */
856          if (delta_cpu->ic_tot < delta_cpu->ic_intrs)
857              delta_cpu->ic_tot = delta_cpu->ic_intrs;

859          delta_cpu->ic_intr_load =
860              delta_cpu->ic_intrs / delta_cpu->ic_tot;
861          intrload += delta_cpu->ic_intr_load;
862          intrnsec += delta_cpu->ic_intrs;

864          cpus++;
865      }

867      if (cpus > 0) {
868          delta->id_avgintrload = intrload / cpus;
869          delta->id_avgintrnsec = intrnsec / cpus;
870      } else {
871          delta->id_avgintrload = 0;
872          delta->id_avgintrnsec = 0;
873      }

875      return (delta);
876  }

878  /*
879  * Compress deltas.
880  */
881  static ib_delta_t *
882  ib_delta_compress(list_t *deltas)
883  {
884      ib_cpu_t      *icpu;
885      ib_ivec_t      *ivec;
886      ib_delta_t      *new_delta, *delta;
887      processorid_t    cpus = 0;
888      int             high_intrload = 0;
889      int             intrs = 0, tot;

891      /* Check if empty list of deltas */
892      if (deltas == NULL || list_is_empty(deltas) != 0) {
893          IB_LOG((CE_CONT, "ib_delta_compress: deltas are empty?"));
894          return (NULL);
895      }

897      /* Allocate a new delta structure */
898      new_delta = ib_delta_create();

900      /*
901      * Loop over the deltas in the list.
902      */
903      for (delta = list_head(deltas); delta != NULL;
904          delta = list_next(deltas, delta)) {

906          /* Compressing bad delta? */
907          if (delta->id_missing) {
908              IB_LOG((CE_CONT,
909                  "ib_delta_compress: compressing bad deltas?"));
910              return (NULL);
911          }

913          FOREACH_CPU(icpu, delta->id_cpu_list) {
914              ib_cpu_t      *new_cpu;
915              ib_ivec_t      *new_ivec;

917              intrs += icpu->ic_intrs;
918              tot += icpu->ic_tot;
919              new_cpu = ib_cpu_create();

```

```

920         new_cpu->ic_cpu_id = icpu->ic_cpu_id;
921         new_cpu->ic_intrs = icpu->ic_intrs;
922         new_cpu->ic_tot = icpu->ic_tot;

924         /* XXX: exists ivecs */
925         FOREACH_IVEC(new_ivec, icpu->ic_ivec_list) {
926             ib_ivec_t *new_delta_ivec;

928             new_delta_ivec = ib_ivec_create(
929                 new_ivec->ii_buspath, new_ivec->ii_ino);

931         }
932     }
933 }

935 FOREACH_CPU(icpu, new_delta->id_cpu_list) {
936     int        begintr = 0;

938     cpus++;

940     FOREACH_IVEC(ivec, icpu->ic_ivec_list) {
941         if (ivec->ii_time > begintr)
942             begintr = ivec->ii_time;
943     }

945     icpu->ic_begintr = begintr;
946     icpu->ic_intr_load = icpu->ic_intrs / icpu->ic_tot;

948     if (high_intrload < icpu->ic_intr_load)
949         high_intrload = icpu->ic_intr_load;

951     if (icpu->ic_tot <= 0)
952         icpu->ic_tot = 100;
953 }

955 if (cpus > 0) {
956     new_delta->id_avgintrload = intrs / tot;
957     new_delta->id_avgintrnsec = intrs / cpus;
958 } else {
959     new_delta->id_avgintrload = 0;
960     new_delta->id_avgintrnsec = 0;
961 }

963 /* XXX: global sleeptime */

965 return (new_delta);
966 }

968 /*
969  * Decide if the load is out of balance.
970  */
971 static int
972 ib_imbalanced(int goodness, int baseline)
973 {
974     if (goodness > 50)
975         return (100);

977     /* XXX: abs */
978     if ((goodness - baseline) > goodness_mindelta)
979         return (100);

981     return (0);
982 }

984 /*
985  * Calculate goodness of a CPU.

```

```

986  */
987 static int
988 ib_goodness_cpu(ib_cpu_t *icpu, int avg_interrupt_load)
989 {
990     int        goodness;
991     int        load, load_no_begintr;

993     load = icpu->ic_intrs / icpu->ic_tot;
994     if (load < avg_interrupt_load)
995         return (0);

997     load_no_begintr = (icpu->ic_intrs - icpu->ic_begintr) / icpu->ic_tot;

999     if ((load > goodness_unsafe_load) && (icpu->ic_num_ivecs > 1))
1000         return (1);

1002     goodness = load - avg_interrupt_load;
1003     if (goodness > load_no_begintr)
1004         goodness = load_no_begintr;

1006     return (goodness);
1007 }

1009 /*
1010  * Calculate goodness.
1011  */
1012 static int
1013 ib_goodness(ib_delta_t *delta)
1014 {
1015     ib_cpu_t    *icpu;
1016     int          goodness, high_goodness = 0;

1018     if (delta->id_missing > 0)
1019         return (1);

1021     FOREACH_CPU(icpu, delta->id_cpu_list) {
1022         goodness = ib_goodness_cpu(icpu, delta->id_avgintrload);
1023         if (!(goodness >= 0 && goodness <= 100)) {
1024             IB_LOG(CE_CONT,
1025                 "ib goodness: cpu goodness out of range?");
1026             return (100);
1027         }

1029         if (goodness == 100)
1030             return (100);

1032         if (goodness > high_goodness)
1033             high_goodness = goodness;
1034     }

1036     return (high_goodness);
1037 }

1039 static void
1040 ib_do_find_goal(list_t ivecs, list_t loads, int goal, int idx)
1041 {
1042     list_t      goals_with;
1043     list_t      goals_without;
1044     int         with, without;
1045     int         which, load;

1048     if (goal <= load) {
1049         with = load;
1050     } else {
1051         /* XXX: do_find_goal */

```

```

1052         with += load;
1053     }
1054
1055     IB_LOG((CE_CONT, "XXX"));
1056
1057     if (with >= goal && without < goal) {
1058         which = 0;
1059     } else if (with < goal && without >= goal) {
1060         which = 1;
1061     } else if (with >= goal && without >= goal) {
1062         which = without < with;
1063     } else {
1064         which = without > with;
1065     }
1066
1067     if (which == 1) {
1068         IB_LOG((CE_CONT, "ib_do_find_goal: going without"));
1069         /* XXX */
1070     } else {
1071         IB_LOG((CE_CONT, "ib_do_find_goal: going with"));
1072         /* XXX */
1073     }
1074 }
1075
1076 typedef struct _ib_goal {
1077     list_node_t    *ig_link;
1078     int            ig_value;
1079 } ib_goal_t;
1080
1081 typedef struct _ib_goal_load {
1082     list_node_t    *igl_link;
1083     int            igl_value;
1084 } ib_goal_load_t;
1085
1086 static void
1087 ib_find_goal(list_t ivecs, int goal)
1088 {
1089     ib_ivec_t      *ivec;
1090     list_t          goals;
1091     int             load;
1092
1093     if (goal <= 0) {
1094         list_create(&goals, sizeof (ib_goal_t),
1095             offsetof (ib_goal_t, ig_link));
1096     } else {
1097         list_t      loads;
1098         hrtime_t    tot = 0;
1099
1100         IB_LOG((CE_CONT, "ib_find_goal: finding goal from intrs XXX"));
1101
1102         FOREACH_IVEC(ivec, ivecs) {
1103             tot += ivec->ii_time;
1104         }
1105
1106         list_create(&loads, sizeof (ib_goal_load_t),
1107             offsetof (ib_goal_load_t, igl_link));
1108
1109         FOREACH_IVEC(ivec, ivecs) {
1110             ib_goal_load_t *igl = kmem_alloc(sizeof (ib_goal_load_t));
1111             igl->igl_value = tot;
1112             list_insert_tail(&loads, igl);
1113
1114             tot -= ivec->ii_time;
1115         }
1116     }
1117 }

```

```

1118 }
1119
1120 static void
1121 ib_do_reconfig_cpu2cpu(ib_delta_t *delta, processorid_t src_cpuid,
1122     processorid_t tgt_cpuid, int src_load)
1123 {
1124     ib_cpu_t      *src_cpu, *tgt_cpu;
1125     ib_ivec_t      *ivec;
1126     list_t         ivecs;
1127     int            goal;
1128     int            avg_nsec;
1129
1130     if (delta == NULL)
1131         return;
1132
1133     goal = delta->id_avgintrnsec;
1134
1135     src_cpu = ib_cpu_find(delta->id_cpu_list, src_cpuid);
1136     if (src_cpu == NULL)
1137         return;
1138
1139     tgt_cpu = ib_cpu_find(delta->id_cpu_list, tgt_cpuid);
1140     if (tgt_cpu == NULL)
1141         return;
1142
1143     avg_nsec = (src_cpu->ic_intrs + tgt_cpu->ic_intrs) / 2;
1144     if (goal < avg_nsec)
1145         goal = avg_nsec;
1146
1147     /*
1148      * Sort interrupt vectors by time.
1149      */
1150     list_create(&ivecs, sizeof (ib_ivec_t),
1151         offsetof (ib_ivec_t, ii_next));
1152
1153     ivec = list_head(&ivecs);
1154     if (ivec->ii_orig_cpu == src_cpuid) {
1155         IB_LOG((CE_CONT, "Keeping XXX on %d",
1156             src_cpuid)); /* ivec->ii_inum, */
1157         goal -= ivec->ii_time;
1158         /* XXX: shift */
1159     }
1160
1161     IB_LOG((CE_CONT, "ib_reconfig_cpu2cpu: inums should total %d", goal));
1162
1163     ib_find_goal(ivecs, goal);
1164
1165 }
1166
1167 static void
1168 ib_do_reconfig_cpu(ib_delta_t *delta, list_t *cpu_sorted_list,
1169     processorid_t old_cpu_id)
1170 {
1171     ib_cpu_t      *icpu;
1172     int            avgintrload;
1173
1174     if (delta == NULL)
1175         return;
1176
1177     icpu = ib_cpu_find(delta->id_cpu_list, old_cpu_id);
1178     if (icpu == NULL)
1179         return;
1180
1181     avgintrload = delta->id_avgintrload;
1182
1183 }

```

```

1185 /*
1186  * Reconfigure interrupt distribution among CPUs.
1187  */
1188 static int
1189 ib_do_reconfig(ib_delta_t *delta)
1190 {
1191     ib_cpu_t      *icpu;
1192     ib_ivec_t      *ivec;
1193     list_t         cpu_sorted_list;
1194     int            goodness, new_goodness;
1195     int            warned = 0;
1196     int            rval = 1, ret = 1;
1197
1198     if (delta == NULL)
1199         return (-1);
1200
1201     goodness = delta->id_goodness;
1202     if (goodness < goodness_mindelta) {
1203         IB_LOG((CE_CONT, "ib_do_reconfig: goodness is good enough"));
1204         return (0);
1205     }
1206
1207     IB_LOG((CE_CONT, "ib_do_reconfig: optimizing interrupt assignments"));
1208
1209     if (delta->id_missing != 0) {
1210         IB_LOG((CE_CONT, "ib_do_reconfig: aborted"));
1211         return (-1);
1212     }
1213
1214     FOREACH_CPU(icpu, delta->id_cpu_list) {
1215         FOREACH_IVEC(ivec, icpu->ic_ivec_list) {
1216             ivec->ii_orig_cpu = icpu->ic_cpu_id;
1217             ivec->ii_now_cpu = icpu->ic_cpu_id;
1218             /* XXX: inum */
1219         }
1220     }
1221
1222     list_create(&cpu_sorted_list, sizeof (ib_cpu_t),
1223               offsetof(ib_cpu_t, ic_next));
1224
1225     /*
1226      * Have we an improvement?
1227      */
1228     new_goodness = ib_goodness(delta);
1229     if (!(new_goodness <= goodness)) {
1230         IB_LOG((CE_CONT,
1231               "ib_do_reconfig: result has worse goodness"));
1232     }
1233
1234     if ((goodness != 100 || new_goodness == 100) &&
1235         goodness - new_goodness < goodness_mindelta) {
1236         IB_LOG((CE_CONT,
1237               "ib_do_reconfig: goodness already near optimum"));
1238         return (0);
1239     }
1240
1241     /*
1242      * Move interrupts.
1243      */
1244
1245     FOREACH_CPU(icpu, delta->id_cpu_list) {
1246         FOREACH_IVEC(ivec, icpu->ic_ivec_list) {
1247             int            error;
1248
1249             if (ivec->ii_orig_cpu == icpu->ic_cpu_id)

```

```

1250             continue;
1251
1252             error = ib_interrupt_do_move(ivec, icpu->ic_cpu_id);
1253             if (error != 0) {
1254                 if (warned++ == 0) {
1255                     IB_LOG((CE_CONT, "ib_do_reconfig: "
1256                           "unable to move interrupt"));
1257                 }
1258
1259                 IB_LOG((CE_CONT, "ib_do_reconfig: "
1260                           "unable to move buspath"));
1261
1262                 ret = -1;
1263             }
1264         }
1265     }
1266
1267     return (rval);
1268 }
1269
1270
1271 /*
1272  * Check if the interrupt load did decrease.
1273  */
1274 static void
1275 ib_interrupt_move_check(ib_delta_t *delta, processorid_t old_cpuid,
1276                       processorid_t new_cpuid)
1277 {
1278     ib_cpu_t      *old_cpu, *new_cpu;
1279
1280     /*
1281      * Check old CPU.
1282      */
1283     old_cpu = ib_cpu_find(delta->id_cpu_list, old_cpuid);
1284     if (old_cpu == NULL)
1285         return;
1286     if (!(old_cpu->ic_tot >= old_cpu->ic_intrs)) {
1287         IB_LOG((CE_CONT,
1288               "Moved interrupts left 100+%% load on source CPU"));
1289     }
1290
1291     /*
1292      * Check new CPU.
1293      */
1294     new_cpu = ib_cpu_find(delta->id_cpu_list, new_cpuid);
1295     if (new_cpu == NULL)
1296         return;
1297     if (!(new_cpu->ic_tot >= new_cpu->ic_intrs)) {
1298         IB_LOG((CE_CONT,
1299               "Moved interrupts left 100+%% load on target CPU"));
1300     }
1301 }
1302
1303 /*
1304  * Actually moving the interrupt.
1305  */
1306 static int
1307 ib_interrupt_do_move(ib_ivec_t *ivec, processorid_t cpu_id)
1308 {
1309     int            ret, result;
1310
1311     struct psm_ops *pops;
1312
1313     //pops = mach_set[0];
1314
1315     //    ret = (*psm_intr_ops)(NULL, &info_hdl, PSM_INTR_OP_SET_CPU,

```

```

1316         //      &result);
1318         return (-1);
1319     }

1321     /*
1322     * Move an interrupt to a different CPU.
1323     */
1324     static int
1325     ib_interrupt_move(ib_delta_t *delta, uint64_t inum, processorid_t old_cpuid,
1326                     processorid_t new_cpuid)
1327     {
1328         ib_cpu_t      *old_cpu, *new_cpu;
1329         ib_ivec_t      *ivec;

1331         if (delta == NULL)
1332             return (-1);

1334         /*
1335         * Remove interrupt vector from old CPU.
1336         */
1337         old_cpu = ib_cpu_find(delta->id_cpu_list, old_cpuid);
1338         if (old_cpu == NULL)
1339             return (-1);

1341         ivec = ib_ivec_find_ino(old_cpu->ic_ivec_list, inum);

1343         old_cpu->ic_intrs -= ivec->ii_time;
1344         old_cpu->ic_intr_load = old_cpu->ic_intrs / old_cpu->ic_tot;
1345         ib_ivec_delete_ino(old_cpu->ic_ivec_list, inum);

1347         /*
1348         * Verify interrupts.
1349         */
1350         if (!(old_cpu->ic_intrs >= 0)) {
1351             IB_LOG((CE_CONT,
1352                  "ib_interrupt_move: interrupt time > total time?"));
1353         }

1355         if (!(ivec->ii_time <= old_cpu->ic_bigintr)) {
1356             IB_LOG((CE_CONT,
1357                  "ib_interrupt_move: interrupt time > big interrupt?"));
1358         }

1360         if (ivec->ii_time >= old_cpu->ic_bigintr) {
1361             ib_ivec_t      *time_ivec;
1362             uint64_t        bigtime = 0;

1364             FOREACH_IVEC(time_ivec, old_cpu->ic_ivec_list) {
1365                 if (time_ivec->ii_time > bigtime)
1366                     bigtime = time_ivec->ii_time;
1367             }
1368         }

1371         /*
1372         * Insert interrupt vector into new CPU.
1373         */
1374         new_cpu = ib_cpu_find(delta->id_cpu_list, new_cpuid);
1375         if (new_cpu == NULL)
1376             return (-1);

1378         ivec->ii_now_cpu = new_cpuid;
1379         new_cpu->ic_intrs += ivec->ii_time;
1380         new_cpu->ic_intr_load = new_cpu->ic_intrs / new_cpu->ic_tot;
1381         ib_ivec_add_ino(new_cpu->ic_ivec_list, ivec);

```

```

1383         if (ivec->ii_time > new_cpu->ic_bigintr)
1384             new_cpu->ic_bigintr = ivec->ii_time;

1386         return (0);
1387     }
1388 #endif /* ! codereview */

```

new/usr/src/uts/common/os/main.c

1

17325 Sat Sep 14 15:00:33 2013

new/usr/src/uts/common/os/main.c

XXX Rearchitect and replace interrupt distribution

```
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21
22 /*
23  * Copyright (c) 1988, 2010, Oracle and/or its affiliates. All rights reserved.
24 */
25
26 /*      Copyright (c) 1988 AT&T */
27 /*      All Rights Reserved      */
28
29 /*
30  * Copyright (c) 2013, Joyent, Inc. All rights reserved.
31 */
32
33 #include <sys/types.h>
34 #include <sys/param.h>
35 #include <sys/symmacros.h>
36 #include <sys/pcb.h>
37 #include <sys/systm.h>
38 #include <sys/signal.h>
39 #include <sys/cred.h>
40 #include <sys/user.h>
41 #include <sys/vfs.h>
42 #include <sys/vnode.h>
43 #include <sys/proc.h>
44 #include <sys/time.h>
45 #include <sys/file.h>
46 #include <sys/priocntl.h>
47 #include <sys/procset.h>
48 #include <sys/disp.h>
49 #include <sys/callo.h>
50 #include <sys/callb.h>
51 #include <sys/debug.h>
52 #include <sys/conf.h>
53 #include <sys/bootconf.h>
54 #include <sys/utsname.h>
55 #include <sys/cmn_err.h>
56 #include <sys/vmparam.h>
57 #include <sys/modctl.h>
58 #include <sys/vm.h>
59 #include <sys/callb.h>
60 #include <sys/ddi_periodic.h>
61 #include <sys/kmem.h>
```

new/usr/src/uts/common/os/main.c

2

```
62 #include <sys/vmem.h>
63 #include <sys/cpuvar.h>
64 #include <sys/cladm.h>
65 #include <sys/corectl.h>
66 #include <sys/exec.h>
67 #include <sys/syscall.h>
68 #include <sys/reboot.h>
69 #include <sys/task.h>
70 #include <sys/exacct.h>
71 #include <sys/autoconf.h>
72 #include <sys/errorq.h>
73 #include <sys/class.h>
74 #include <sys/stack.h>
75 #include <sys/brand.h>
76 #include <sys/mmapobj.h>
77
78 #include <vm/as.h>
79 #include <vm/seg_kmem.h>
80 #include <sys/dc_ki.h>
81
82 #include <c2/audit.h>
83 #include <sys/bootprops.h>
84
85 /* well known processes */
86 proc_t *proc_sched;          /* memory scheduler */
87 proc_t *proc_init;          /* init */
88 proc_t *proc_pageout;        /* pageout daemon */
89 proc_t *proc_fsflush;        /* fsflush daemon */
90 proc_t *proc_intrd;          /* interrupt daemon */
91 #endif /* ! codereview */
92
93 pgcnt_t maxmem;              /* Maximum available memory in pages. */
94 pgcnt_t freemem;             /* Current available memory in pages. */
95 int interrupts_unleashed;     /* set when we do the first spl0() */
96
97 kmem_cache_t *process_cache; /* kmem cache for proc structures */
98
99 /*
100  * Indicates whether the auditing module (c2audit) is loaded. Possible
101  * values are:
102  * 0 - c2audit module is excluded in /etc/system and cannot be loaded
103  * 1 - c2audit module is not loaded but can be anytime
104  * 2 - c2audit module is loaded
105  */
106 int audit_active = C2AUDIT_DISABLED;
107
108 /*
109  * Process 0's lwp directory and lwpid hash table.
110  */
111 lwpdir_t p0_lwpdir[2];
112 tidhash_t p0_tidhash[2];
113 lwpent_t p0_lep;
114
115 /*
116  * Machine-independent initialization code
117  * Called from cold start routine as
118  * soon as a stack and segmentation
119  * have been established.
120  * Functions:
121  * clear and free user core
122  * turn on clock
123  * hand craft 0th process
124  * call all initialization routines
125  * fork      - process 0 to schedule
126  *           - process 1 execute bootstrap
127  *           - process 2 to page out
```

```

128 *      create system threads
129 */

131 int cluster_bootflags = 0;

133 void
134 cluster_wrapper(void)
135 {
136     cluster();
137     panic("cluster() returned");
138 }

140 char initname[INITNAME_SZ] = "/sbin/init"; /* also referenced by zone0 */
141 char initargs[BOOTARGS_MAX] = ""; /* also referenced by zone0 */

143 /*
144  * Construct a stack for init containing the arguments to it, then
145  * pass control to exec_common.
146  */
147 int
148 exec_init(const char *initpath, const char *args)
149 {
150     caddr32_t ucp;
151     caddr32_t *uap;
152     caddr32_t *argv;
153     caddr32_t exec_fnamep;
154     char *scratchargs;
155     int i, sarg;
156     size_t argvlen, alen;
157     boolean_t in_arg;
158     int argc = 0;
159     int error = 0, count = 0;
160     proc_t *p = ttproc(curthread);
161     klpw_t *lwp = ttolwp(curthread);
162     int brand_action;

164     if (args == NULL)
165         args = "";

167     alen = strlen(initpath) + 1 + strlen(args) + 1;
168     scratchargs = kmem_alloc(alen, KM_SLEEP);
169     (void) snprintf(scratchargs, alen, "%s %s", initpath, args);

171     /*
172      * We do a quick two state parse of the string to sort out how big
173      * argc should be.
174      */
175     in_arg = B_FALSE;
176     for (i = 0; i < strlen(scratchargs); i++) {
177         if (scratchargs[i] == ' ' || scratchargs[i] == '\0') {
178             if (in_arg) {
179                 in_arg = B_FALSE;
180                 argc++;
181             }
182         } else {
183             in_arg = B_TRUE;
184         }
185     }
186     argvlen = sizeof (caddr32_t) * (argc + 1);
187     argv = kmem_zalloc(argvlen, KM_SLEEP);

189     /*
190      * We pull off a bit of a hack here. We work our way through the
191      * args string, putting nulls at the ends of space delimited tokens
192      * (boot args don't support quoting at this time). Then we just
193      * copy the whole mess to userland in one go. In other words, we

```

```

194 *      transform this: "init -s -r\0" into this on the stack:
195 *
196 *      -0x00 \0
197 *      -0x01 r
198 *      -0x02 - <-----
199 *      -0x03 \0
200 *      -0x04 s
201 *      -0x05 - <-----
202 *      -0x06 \0
203 *      -0x07 t
204 *      -0x08 i
205 *      -0x09 n
206 *      -0x0a i <-----
207 *      -0x10 NULL (argv[3])
208 *      -0x14 ----- (argv[2])
209 *      -0x18 ----- (argv[1])
210 *      -0x1c ----- (argv[0])
211 *
212 * Since we know the value of ucp at the beginning of this process,
213 * we can trivially compute the argv[] array which we also need to
214 * place in userland: argv[i] = ucp - sarg(i), where ucp is the
215 * stack ptr, and sarg is the string index of the start of the
216 * argument.
217 */
218 ucp = (caddr32_t)(uintptr_t)p->p_usrstack;

220 argc = 0;
221 in_arg = B_FALSE;
222 sarg = 0;

224 for (i = 0; i < alen; i++) {
225     if (scratchargs[i] == ' ' || scratchargs[i] == '\0') {
226         if (in_arg == B_TRUE) {
227             in_arg = B_FALSE;
228             scratchargs[i] = '\0';
229             argv[argc++] = ucp - (alen - sarg);
230         }
231         } else if (in_arg == B_FALSE) {
232             in_arg = B_TRUE;
233             sarg = i;
234         }
235     }
236     ucp -= alen;
237     error |= copyout(scratchargs, (caddr_t)(uintptr_t)ucp, alen);

239     uap = (caddr32_t *)P2ALIGN((uintptr_t)ucp, sizeof (caddr32_t));
240     uap--; /* advance to be below the word we're in */
241     uap -= (argc + 1); /* advance argc words down, plus one for NULL */
242     error |= copyout(argv, uap, argvlen);

244     if (error != 0) {
245         zcmn_err(p->p_zone->zone_id, CE_WARN,
246             "Could not construct stack for init.\n");
247         kmem_free(argv, argvlen);
248         kmem_free(scratchargs, alen);
249         return (EFAULT);
250     }

252     exec_fnamep = argv[0];
253     kmem_free(argv, argvlen);
254     kmem_free(scratchargs, alen);

256     /*
257      * Point at the arguments.
258      */
259     lwp->lwp_ap = lwp->lwp_arg;

```



```

260     lwp->lwp_arg[0] = (uintptr_t)exec_fnamep;
261     lwp->lwp_arg[1] = (uintptr_t)uap;
262     lwp->lwp_arg[2] = NULL;
263     curthread->t_post_sys = 1;
264     curthread->t_sysnum = SYS_execve;

266     /*
267      * If we are executing init from zsched, we may have inherited its
268      * parent process's signal mask. Clear it now so that we behave in
269      * the same way as when started from the global zone.
270      */
271     sigemptyset(&curthread->t_hold);

273     brand_action = ZONE_IS_BRANDED(p->p_zone) ? EBA_BRAND : EBA_NONE;
274 again: error = exec_common((const char *) (uintptr_t)exec_fnamep,
275                          (const char **) (uintptr_t)uap, NULL, brand_action);
276
278     /*
279      * Normally we would just set lwp_argsaved and t_post_sys and
280      * let post_syscall reset lwp_ap for us. Unfortunately,
281      * exec_init isn't always called from a system call. Instead
282      * of making a mess of trap_cleanup, we just reset the args
283      * pointer here.
284      */
285     reset_syscall_args();

287     switch (error) {
288     case 0:
289         return (0);

291     case ENOENT:
292         zcmn_err(p->p_zone->zone_id, CE_WARN,
293                 "exec(%s) failed (file not found).\n", initpath);
294         return (ENOENT);

296     case EAGAIN:
297     case EINTR:
298         ++count;
299         if (count < 5) {
300             zcmn_err(p->p_zone->zone_id, CE_WARN,
301                     "exec(%s) failed with errno %d. Retrying...\n",
302                     initpath, error);
303             goto again;
304         }
305     }

307     zcmn_err(p->p_zone->zone_id, CE_WARN,
308             "exec(%s) failed with errno %d.", initpath, error);
309     return (error);
310 }

312 /*
313  * This routine does all of the common setup for invoking init; global
314  * and non-global zones employ this routine for the functionality which is
315  * in common.
316  *
317  * This program (init, presumably) must be a 32-bit process.
318  */
319 int
320 start_init_common()
321 {
322     proc_t *p = curproc;
323     ASSERT_STACK_ALIGNED();
324     p->p_zone->zone_proc_initpid = p->p_pid;

```

```

326     p->p_cstime = p->p_stime = p->p_cutime = p->p_utime = 0;
327     p->p_usrstack = (caddr_t)USRSTACK32;
328     p->p_model = DATAMODEL_ILP32;
329     p->p_stkprot = PROT_ZFOD & ~PROT_EXEC;
330     p->p_datprot = PROT_ZFOD & ~PROT_EXEC;
331     p->p_stk_ctl = INT32_MAX;

333     p->p_as = as_alloc();
334     p->p_as->a_proc = p;
335     p->p_as->a_userlimit = (caddr_t)USERLIMIT32;
336     (void) hat_setup(p->p_as->a_hat, HAT_INIT);

338     init_core();

340     init_mstate(curthread, LMS_SYSTEM);
341     return (exec_init(p->p_zone->zone_initname, p->p_zone->zone_bootargs));
342 }

344 /*
345  * Start the initial user process for the global zone; once running, if
346  * init should subsequently fail, it will be automatically be caught in the
347  * exit(2) path, and restarted by restart_init().
348  */
349 static void
350 start_init(void)
351 {
352     proc_init = curproc;

354     ASSERT(curproc->p_zone->zone_initname != NULL);

356     if (start_init_common() != 0)
357         halt("unix: Could not start init");
358     lwp_rtt();
359 }

361 void
362 main(void)
363 {
364     proc_t      *p = ttoproc(curthread);      /* &p0 */
365     int          (*initptr)();
366     extern void  sched();
367     extern void  fsflush();
368     extern void  interrupt_balancer();
369 #endif /* ! codereview */
370     extern int   (*init_tbl[])();
371     extern int   (*mp_init_tbl[])();
372     extern id_t  syscid, defaultcid;
373     extern int   swaploaded;
374     extern int   netboot;
375     extern ib_boot_prop_t *iscsiboot_prop;
376     extern void  vm_init(void);
377     extern void  cbe_init_pre(void);
378     extern void  cbe_init(void);
379     extern void  clock_tick_init_pre(void);
380     extern void  clock_tick_init_post(void);
381     extern void  clock_init(void);
382     extern void  physio_bufs_init(void);
383     extern void  pm_cfb_setup_intr(void);
384     extern int   pm_adjust_timestamps(dev_info_t *, void *);
385     extern void  start_other_cpus(int);
386     extern void  sysevent_evt_thrinit();
387     extern kmutex_t ualock;
388 #if defined(__x86)
389     extern void  fastboot_post_startup(void);
390     extern void  progressbar_start(void);
391 #endif

```

```

392 /*
393  * In the horrible world of x86 in-lines, you can't get symbolic
394  * structure offsets a la genassym. This assertion is here so
395  * that the next poor slob who innocently changes the offset of
396  * cpu_thread doesn't waste as much time as I just did finding
397  * out that it's hard-coded in i86/ml/i86.il. Similarly for
398  * curcpup. You're welcome.
399  */
400 ASSERT(CPU == CPU->cpu_self);
401 ASSERT(curthread == CPU->cpu_thread);
402 ASSERT_STACK_ALIGNED();

404 /*
405  * We take the ualock until we have completed the startup
406  * to prevent kadmin() from disrupting this work. In particular,
407  * we don't want kadmin() to bring the system down while we are
408  * trying to start it up.
409  */
410 mutex_enter(&ualock);

412 /*
413  * Setup root lgroup and leaf lgroup for CPU 0
414  */
415 lgrp_init(LGRP_INIT_STAGE2);

417 /*
418  * Once 'startup()' completes, the thread_reaper() daemon would be
419  * created(in thread_init()). After that, it is safe to create threads
420  * that could exit. These exited threads will get reaped.
421  */
422 startup();
423 segkmem_gc();
424 callb_init();
425 cbe_init_pre(); /* x86 must initialize gethrtimef before timer_init */
426 ddi_periodic_init();
427 cbe_init();
428 callout_init(); /* callout table MUST be init'd after cyclics */
429 clock_tick_init_pre();
430 clock_init();

432 #if defined(__x86)
433 /*
434  * The progressbar thread uses cv_reltimedwait() and hence needs to be
435  * started after the callout mechanism has been initialized.
436  */
437 progressbar_start();
438 #endif

439 /*
440  * On some platforms, clkinitf() changes the timing source that
441  * gethrtime_unscaled() uses to generate timestamps. cbe_init() calls
442  * clkinitf(), so re-initialize the microstate counters after the
443  * timesource has been chosen.
444  */
445 init_mstate(&t0, LMS_SYSTEM);
446 init_cpu_mstate(CPU, CMS_SYSTEM);

448 /*
449  * May need to probe to determine latencies from CPU 0 after
450  * gethrtime() comes alive in cbe_init() and before enabling interrupts
451  * and copy and release any temporary memory allocated with BOP_ALLOC()
452  * before release_bootstrap() frees boot memory
453  */
454 lgrp_init(LGRP_INIT_STAGE3);

456 /*
457  * Call all system initialization functions.

```

```

458 */
459 for (initptr = &init_tbl[0]; *initptr; initptr++)
460     (**initptr)();
461 /*
462  * Load iSCSI boot properties
463  */
464 ld_ib_prop();
465 /*
466  * initialize vm related stuff.
467  */
468 vm_init();

470 /*
471  * initialize buffer pool for raw I/O requests
472  */
473 physio_bufs_init();

475 ttolwp(curthread)->lwp_error = 0; /* XXX kludge for SCSI driver */

477 /*
478  * Drop the interrupt level and allow interrupts. At this point
479  * the DDI guarantees that interrupts are enabled.
480  */
481 (void) spl0();
482 interrupts_unleashed = 1;

484 /*
485  * Create kmem cache for proc structures
486  */
487 process_cache = kmem_cache_create("process_cache", sizeof (proc_t),
488     0, NULL, NULL, NULL, NULL, NULL, 0);

490 vfs_mountroot(); /* Mount the root file system */
491 errorq_init(); /* after vfs_mountroot() so DDI root is ready */
492 cpu_kstat_init(CPU); /* after vfs_mountroot() so TOD is valid */
493 ddi_walk_devs(ddi_root_node(), pm_adjust_timestamps, NULL);
494 /* after vfs_mountroot() so hrestime is valid */

496 post_startup();
497 swaploaded = 1;

499 /*
500  * Initialize Solaris Audit Subsystem
501  */
502 audit_init();

504 /*
505  * Plumb the protocol modules and drivers only if we are not
506  * networked booted, in this case we already did it in rootconf().
507  */
508 if (netboot == 0 && iscsiboot_prop == NULL)
509     (void) strplumb();

511 gethrtime(&PTOU(curproc)->u_start);
512 curthread->t_start = PTOU(curproc)->u_start.tv_sec;
513 p->p_mstart = gethrtime();

515 /*
516  * Perform setup functions that can only be done after root
517  * and swap have been set up.
518  */
519 consconfig();
520 #ifndef __sparc
521 release_bootstrap();
522 #endif

```

```

524 /*
525  * attach drivers with ddi-forceattach prop
526  * It must be done early enough to load hotplug drivers (e.g.
527  * pcmcia nexus) so that devices enumerated via hotplug is
528  * available before I/O subsystem is fully initialized.
529  */
530 i_ddi_forceattach_drivers();

532 /*
533  * Set the scan rate and other parameters of the paging subsystem.
534  */
535 setupclock(0);

537 /*
538  * Initialize process 0's lwp directory and lwpid hash table.
539  */
540 p->p_lwpdir = p->p_lwpfree = p0_lwpdir;
541 p->p_lwpdir->ld_next = p->p_lwpdir + 1;
542 p->p_lwpdir_sz = 2;
543 p->p_tidhash = p0_tidhash;
544 p->p_tidhash_sz = 2;
545 p0_lep.le_thread = curthread;
546 p0_lep.le_lwpid = curthread->t_tid;
547 p0_lep.le_start = curthread->t_start;
548 lwp_hash_in(p, &p0_lep, p0_tidhash, 2, 0);

550 /*
551  * Initialize extended accounting.
552  */
553 exacct_init();

555 /*
556  * Initialize threads of sysevent event channels
557  */
558 sysevent_evc_thrinit();

560 /*
561  * This must be done after post_startup() but before
562  * start_other_cpus()
563  */
564 lgrp_init(LGRP_INIT_STAGE4);

566 /*
567  * Perform MP initialization, if any.
568  */
569 start_other_cpus(0);

571 #ifdef __sparc
572 /*
573  * Release bootstrap here since PROM interfaces are
574  * used to start other CPUs above.
575  */
576 release_bootstrap();
577 #endif

579 /*
580  * Finish lgrp initialization after all CPUs are brought online.
581  */
582 lgrp_init(LGRP_INIT_STAGE5);

584 /*
585  * After mp_init(), number of cpus are known (this is
586  * true for the time being, when there are actually
587  * hot pluggable cpus then this scheme would not do).
588  * Any per cpu initialization is done here.
589  */

```

```

590 kmem_mp_init();
591 vmem_update(NULL);

593 clock_tick_init_post();

595 for (initptr = &mp_init_tbl[0]; *initptr; initptr++)
596     (**initptr)();

598 /*
599  * These must be called after start_other_cpus
600  */
601 pm_cfb_setup_intr();
602 #if defined(__x86)
603 fastboot_post_startup();
604 #endif

606 /*
607  * Make init process; enter scheduling loop with system process.
608  */
609 /* Note that we manually assign the pids for these processes, for
610  * historical reasons. If more pre-assigned pids are needed,
611  * FAMOUS_PIDS will have to be updated.
612  */

614 /* create init process */
615 if (newproc(start_init, NULL, defaultcid, 59, NULL,
616     FAMOUS_PID_INIT))
617     panic("main: unable to fork init.");

619 /* create pageout daemon */
620 if (newproc(pageout, NULL, syscid, maxclsypri - 1, NULL,
621     FAMOUS_PID_PAGEOUT))
622     panic("main: unable to fork pageout()");

624 /* create fsflush daemon */
625 if (newproc(fsflush, NULL, syscid, minclsypri, NULL,
626     FAMOUS_PID_FSFLUSH))
627     panic("main: unable to fork fsflush()");

629 /* create interrupt balancer process */
630 if (newproc(interrupt_balancer, NULL, syscid, minclsypri, NULL, 0))
631     panic("main: unable to fork interrupt_balancer()");

633 #endif /* ! codereview */
634 /* create cluster process if we're a member of one */
635 if (cluster_bootflags & CLUSTER_BOOTED) {
636     if (newproc(cluster_wrapper, NULL, syscid, minclsypri,
637         NULL, 0)) {
638         panic("main: unable to fork cluster()");
639     }
640 }

642 /*
643  * Create system threads (threads are associated with p0)
644  */

646 /* create module uninstall daemon */
647 /* BugID 1132273. If swapping over NFS need a bigger stack */
648 (void) thread_create(NULL, 0, (void (*)())mod_uninstall_daemon,
649     NULL, 0, &p0, TS_RUN, minclsypri);

651 (void) thread_create(NULL, 0, seg_pasync_thread,
652     NULL, 0, &p0, TS_RUN, minclsypri);

654 pid_setmin();

```

```
656      /* system is now ready */
657      mutex_exit(&ualock);

659      bcopy("sched", PTOU(curproc)->u_psargs, 6);
660      bcopy("sched", PTOU(curproc)->u_comm, 5);
661      sched();
662      /* NOTREACHED */
663 }
```

new/usr/src/uts/common/sys/uadmin.h

1

5225 Sat Sep 14 15:00:34 2013

new/usr/src/uts/common/sys/uadmin.h

XXX Rearchitect and replace interrupt distribution

```
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 */

26 /*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
27 /*      All Rights Reserved */

30 #ifndef _SYS_UADMIN_H
31 #define _SYS_UADMIN_H

34 #if !defined(_ASM)
35 #include <sys/types.h>
36 #include <sys/cred.h>
37 #endif

39 #ifdef __cplusplus
40 extern "C" {
41 #endif

43 #define A_REBOOT      1
44 #define A_SHUTDOWN    2
45 #define A_FREEZE      3      /* For freeze and thaw */
46 #define A_REMOUNT     4
47 #define A_DUMP        5
48 #define A_FTRACE      15
49 #define A_SWAPCTL     16
50 /*      17-21      reserved for obsolete interface */
51 #define A_SDTTEST     22      /* DTrace sdt:::test */
52 #define A_CONFIG      23      /* For system configuration */
53 #define A_INTRD       24      /* Interrupt balancer */
54 #endif /* ! codereview */

56 #define AD_UNKNOWN    -1      /* no method */
57 #define AD_HALT       0       /* halt the processor */
58 #define AD_BOOT       1       /* multi-user reboot */
59 #define AD_IBOOT      2       /* multi-user reboot, ask for name of file */
60 #define AD_SBOOT      3       /* single-user reboot */
61 #define AD_SIBOOT     4       /* single-user reboot, ask for name of file */
```

new/usr/src/uts/common/sys/uadmin.h

2

```
62 #define AD_POWEROFF   6       /* software poweroff */
63 #define AD_NOSYNC     7       /* do not sync filesystems on next A_DUMP */
64 #define AD_FASTREBOOT 8       /* bypass firmware and boot loader */
65 #define AD_FASTREBOOT_DRYRUN 9 /* Fast reboot Dry run */

67 /*
68 * Functions reserved for A_FREEZE (may not be available on all platforms)
69 * Note: AD_COMPRESS, AD_CHECK and AD_FORCE are now obsolete
70 * The first two are succeeded by AD_SUSPEND_TO_DISK and
71 * AD_CHECK_SUSPEND_TO_DISK respectively.
72 * AD_FORCE should not be used by any new application
73 *
74 * We maintain compatibility with the earlier interfaces:
75 * AD_COMPRESS and AD_CHECK, by preserving those values
76 * in the corresponding new interfaces
77 */

79 #define AD_COMPRESS    0       /* store state file compressed during CPR */
80 #define AD_FORCE       1       /* force to do AD_COMPRESS */
81 #define AD_CHECK       2       /* test if CPR module is available */
82 #define AD_SUSPEND_TO_DISK AD_COMPRESS /* A_FREEZE, CPR or ACPI S4 */
83 #define AD_CHECK_SUSPEND_TO_DISK AD_CHECK /* A_FREEZE, CPR/S4 capable? */
84 #define AD_SUSPEND_TO_RAM 20    /* A_FREEZE, S3 */
85 #define AD_CHECK_SUSPEND_TO_RAM 21 /* A_FREEZE, S3 capable? */

87 /*
88 * NOTE: the following defines comprise an Unstable interface. Their semantics
89 * may change or they may be removed completely in a later release
90 */
91 #define AD_REUSEINIT   3       /* prepare for AD_REUSABLE */
92 #define AD_REUSABLE    4       /* create reusable statefile */
93 #define AD_REUSEFINI   5       /* revert to normal CPR mode (not reusable) */

95 #define AD_FTRACE_START 1
96 #define AD_FTRACE_STOP 2

98 /*
99 * Functions of A_CONFIG. Unstable interface.
100 */
101 #define AD_UPDATE_BOOT_CONFIG 1 /* Update boot config variables */

103 /*
104 * When 'mdep' (the second argument to uadmin(2)) is initialized for A_REBOOT,
105 * A_SHUTDOWN or A_DUMP, it represents the boot arguments string of at most
106 * 256 characters.
107 */
108 #define BOOTARGS_MAX 256

110 #if !defined(_KERNEL)
111 /*
112 * FMRI for boot-config service.
113 */
114 #define FMRI_BOOT_CONFIG \
115     "svc:/system/boot-config:default"

117 /*
118 * Property group that contains all Fast Reboot configuration properties.
119 */
120 #define BOOT_CONFIG_PG_PARAMS "config"

122 /*
123 * Property group that contains all Fast Reboot blacklisting information.
124 */
125 #define BOOT_CONFIG_PG_FBLACKLIST "fastreboot_blacklist"

127 */
```

```
128 * Non-persistent property group which contains all the properties that
129 * will override settings in the BOOT_CONFIG_PG_PARAMS property group.
130 */
131 #define BOOT_CONFIG_PG_OVR                "config_ovr"

133 #endif /* _KERNEL */

135 /*
136 * Flag representations of fastboot configuration.
137 */
138 #define UA_FASTREBOOT_DEFAULT    0x01
139 #define UA_FASTREBOOT_ONPANIC    0x02

141 #define FASTREBOOT_DEFAULT        "fastreboot_default"
142 #define FASTREBOOT_ONPANIC        "fastreboot_onpanic"
143 #define FASTREBOOT_ONPANIC_CMDLINE "fastreboot_onpanic_cmdline"

145 #define FASTREBOOT_ONPANIC_NOTSET(p) \
146     (strcmp((p), "false") == 0 || \
147      strcmp((p), "no") == 0 || \
148      strcmp((p), "0") == 0)

150 #define FASTREBOOT_ONPANIC_ISSET(p) \
151     (strcmp((p), "true") == 0 || \
152      strcmp((p), "yes") == 0 || \
153      strcmp((p), "1") == 0)

155 #if !defined(_ASM)

157 #if defined(_KERNEL)
158 extern kmutex_t ualock;
159 extern void mdboot(int, int, char *, boolean_t);
160 extern void mdpreboot(int, int, char *);
161 extern int kadmin(int, int, void *, cred_t *);
162 extern void killall(zoneid_t);
163 #endif

165 #if defined(__STDC__)
166 extern int uadmin(int, int, uintptr_t);
167 #else
168 extern int uadmin();
169 #endif

171 #endif /* _ASM */

173 #ifdef __cplusplus
174 }
175 #endif

177 #endif /* _SYS_UADMIN_H */
```