```
**********************************************************
   51457 Sun Dec 14 23:31:12 2014
new/usr/src/cmd/avs/sdbc/scmadm.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 125 void
 126 sdbc_report_error(spcs_s_info_t *ustatus)
 127 {
 128         if (*ustatus != (uintptr_t)NULL) {
 128         if (*ustatus != NULL) {
 129                 spcs_s_report(*ustatus, stderr);
 130                 spcs_s_ufree(ustatus);
 131         } else
 132                 (void) fprintf(stderr, "%s\n", strerror(errno));
 133 }
______unchanged_portion_omitted_
```

****
17685 Sun Dec 14 23:31:13 2014
new/usr/src/cmd/bart/compare.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
****
_____***unchanged_portion_omitted_***

```
 228 static void
 229 parse_line(char *line, char *fname, char *type, char *size, char *mode,
 230 char *acl, char *mtime, char *uid, char *gid, char *contents, char *devnode,
 231 char *dest)
 232 {
 233         int             pos, line_len;

 235         line_len = strlen(line);
 236         pos = 0;

 238         get_token(line, &pos, line_len, fname, PATH_MAX);
 239         get_token(line, &pos, line_len, type, TYPE_SIZE);
 240         get_token(line, &pos, line_len, size, MISC_SIZE);
 241         get_token(line, &pos, line_len, mode, MISC_SIZE);
 242         get_token(line, &pos, line_len, acl, ACL_SIZE);
 243         get_token(line, &pos, line_len, mtime, MISC_SIZE);
 244         get_token(line, &pos, line_len, uid, MISC_SIZE);
 245         get_token(line, &pos, line_len, gid, MISC_SIZE);

 247         /* Reset these fields... */

 249         *contents = '\0';
 249         *contents = NULL;
 250         *devnode = '\0';
 251         *dest = '\0';

 253         /* Handle filetypes which have a last field..... */
 254         if (type[0] == 'F')
 255                 get_token(line, &pos, line_len, contents, PATH_MAX);
 256         else if ((type[0] == 'B') || (type[0] == 'C'))
 257                 get_token(line, &pos, line_len, devnode, PATH_MAX);
 258         else if (type[0] == 'L')
 259                 get_token(line, &pos, line_len, dest, PATH_MAX);
 260 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    5416 Sun Dec 14 23:31:13 2014
new/usr/src/cmd/bnu/getprm.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
```

_____*unchanged_portion_omitted_*

```
 180 /*
 181  * split - split the name into parts:
 182  *      arg  - original string
 183  *      sys  - leading system name
 184  *      fwd  - intermediate destinations, if not NULL, otherwise
 185  *             only split into two parts.
 186  *      file - filename part
 187  */

 189 int
 190 split(arg, sys, fwd, file)
 191 char *arg, *sys, *fwd, *file;
 192 {
 193     wchar_t *cl, *cr, *n;
 194     int retval = 0;
 195     wchar_t     wcbuf[MAXFULLNAME];
 196     wchar_t     tmpbuf[MAXFULLNAME];
 197     wchar_t     myname[MAXFULLNAME];

 199     *sys = *file = NULLCHAR;
 200     if ( fwd != (char *) NULL )
 201         *fwd = NULLCHAR;

 203     /* uux can use parentheses for output file names */
 204     /* we'll check here until  we can move it to uux */
 205     if (EQUALS(Progname,"uux") && (*arg == LQUOTE)) {
 206         char *c;
 207         c = bal(arg++, RQUOTE);
 208         (void) strncpy(file, arg, c-arg);
 209         file[c-arg] = NULLCHAR;
 210         return(retval);
 211         }
 212

 214     mbstowcs(myname, Myname, MAXFULLNAME);
 215     mbstowcs(wcbuf, arg, MAXFULLNAME);
 216     for (n=wcbuf ;; n=cl+1) {
 217         cl = wcschr(n, (wchar_t)'!');
 218         if (cl == NULL) {
 219             /* no ! in n */
 220             (void) wcstombs(file, n, MAXFULLNAME);
 221             return(retval);
 222         }

 224         retval = 1;
 225         if (cl == n)    /* leading ! */
 226             continue;
 227         if (WEQUALSN(myname, n, cl - n) && myname[cl - n] == NULLCHAR)
 228             continue;

 230         (void) wcsncpy(tmpbuf, n, cl-n);
 231         tmpbuf[cl-n] = NULLCHAR;
 232         (void) wcstombs(sys, tmpbuf, MAXFULLNAME);

 234         if (fwd != (char *) NULL) {
 235             if (cl != (cr = wcsrchr(n, (wchar_t)'!'))) {
 236                 /*  more than one ! */
```



```
 237                 wcsncpy(tmpbuf, cl+1, cr-cl-1);
 238                 tmpbuf[cr-cl-1] = L'\0';
 238                 tmpbuf[cr-cl-1] = NULL;
 239                 (void) wcstombs(fwd, tmpbuf, MAXFULLNAME);
 240             }
 241         } else {
 242             cr = cl;
 243         }

 245         (void) wcstombs(file, cr+1, MAXFULLNAME);
 246         return(retval);
 247     }
 248     /*NOTREACHED*/
 249 }
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    3658 Sun Dec 14 23:31:13 2014
new/usr/src/cmd/bnu/gio.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

  65 int
  66 gturnon()
  67 {
  68         struct pack *pkopen();
  69         if (setjmp(Gfailbuf))
  70                 return(FAIL);
  71         gsig=signal(SIGALRM, galarm);
  72         if (Debug > 4)
  73                 pkdebug = 1;
  74         Pk = pkopen(Ifn, Ofn);
  75         if ( Pk == NULL)
  75         if ((int) Pk == NULL)
  76                 return(FAIL);
  77         return(0);
  78 }
______unchanged_portion_omitted_
```

****
10898 Sun Dec 14 23:31:14 2014
new/usr/src/cmd/bnu/interface.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
****
_____***unchanged_portion_omitted_***

```
 275 /*
 276  *      twrite - tli write routine
 277  */
 278 #define N_CHECK 100
 279 static ssize_t
 280 twrite(fd, buf, nbytes)
 281 int             fd;
 282 char            *buf;
 283 unsigned        nbytes;
 284 {
 285         register int            i, ret;
 286         static int              n_writ, got_info;
 287         static struct t_info    info;

 289         if (got_info == 0) {
 290                 if (t_getinfo(fd, &info) != 0) {
 291                         tfaillog(fd, "twrite: t_getinfo\n");
 292                         return (FAIL);
 293                 }
 294                 got_info = 1;
 295         }

 297         /* on every N_CHECKth call, check that are still in DATAXFER state */
 298         if (++n_writ == N_CHECK) {
 299                 n_writ = 0;
 300                 if (t_getstate(fd) != T_DATAXFER)
 301                         return (FAIL);
 302         }

 304         if (info.tsdu <= 0 || nbytes <= info.tsdu)
 305                 return (t_snd(fd, buf, nbytes, 0));
 305                 return (t_snd(fd, buf, nbytes, NULL));

 307         /* if get here, then there is a limit on transmit size  */
 308         /* (info.tsdu > 0) and buf exceeds it                   */
 309         i = ret = 0;
 310         while (nbytes >= info.tsdu) {
 311                 if ((ret = t_snd(fd,  &buf[i], info.tsdu, 0)) != info.tsdu)
 311                 if ((ret = t_snd(fd,  &buf[i], info.tsdu, NULL)) != info.tsdu)
 312                         return ((ret >= 0 ? (i + ret) : ret));
 313                 i += info.tsdu;
 314                 nbytes -= info.tsdu;
 315         }
 316         if (nbytes != 0) {
 317                 if ((ret = t_snd(fd,  &buf[i], nbytes, 0)) != nbytes)
 317                 if ((ret = t_snd(fd,  &buf[i], nbytes, NULL)) != nbytes)
 318                         return ((ssize_t)(ret >= 0 ? (i + ret) : ret));
 319                 i += nbytes;
 320         }
 321         return ((ssize_t)i);
 322 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   18010 Sun Dec 14 23:31:14 2014
new/usr/src/cmd/bnu/sysfiles.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 458 /*
 459  * get next line from Systems file
 460  * return TRUE if successful, FALSE if not
 461  */
 462 GLOBAL int
 463 getsysline(buf, len)
 464 char *buf;
 465 {
 466         char    *prev = _uu_setlocale(LC_ALL, "C");

 468         if (Systems[0] == NULL)
 469                 /* not initialized via setservice() - use default */
 470                 setservice("uucico");

 472         /* initialize devices and dialers whenever a new line is read */
 473         /* from systems */
 474         devreset();
 475         if (fsystems == NULL)
 476                 if (nextsystems() == FALSE) {
 477                         (void) _uu_resetlocale(LC_ALL, prev);
 478                         return(FALSE);
 479                 }

 481         ASSERT(len >= BUFSIZ, "BUFFER TOO SMALL", "getsysline", 0);
 482         for(;;) {
 483                 while (getaline(fsystems, buf) != 0)
 483                 while (getaline(fsystems, buf) != NULL)
 484                     if ((*buf != '#') && (*buf != ' ') &&
 485                         (*buf != '\t') && (*buf != '\n')) {
 486                         (void) _uu_resetlocale(LC_ALL, prev);
 487                         return(TRUE);
 488                 }
 489                 if (nextsystems() == FALSE) {
 490                         (void) _uu_resetlocale(LC_ALL, prev);
 491                         return(FALSE);
 492                 }
 493         }
 494 }
______unchanged_portion_omitted_
```

```
**********************************************************
   13356 Sun Dec 14 23:31:14 2014
new/usr/src/cmd/bnu/uudecode.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 283 /*
 284  * copy from in to out, decoding as you go along.
 285  */

 287 static void
 288 decode(FILE *in, FILE *out, int base64)
 289 {
 290         char    inbuf[120], *ibp, *iptr;
 291         unsigned char   outbuf[BUFSIZE], *obp, *optr;
 292         int     n, octets, warned, endseen, numbase64chars;
 293         unsigned char chr[4], curchr, ch;
 294         longlong_t line;

 296         if (! base64) { /* use historical algorithm */
 297                 warned = 0;
 298                 for (line = 1; ; line++) {
 299                         /* for each input line */
 300                         if (fgets(inbuf, sizeof (inbuf), in) == NULL) {
 301                                 (void) fprintf(stderr,
 302                                     gettext("No end line\n"));
 303                                 exit(5);
 304                         }

 306                         /* Is line == 'end\n'? */
 307                         if (strcmp(inbuf, "end\n") == 0) {
 308                                 break;
 309                         }

 311                         n = DEC(inbuf[0]);

 313                         if (n < 0)
 314                                 continue;

 316                         /*
 317                          * Decode data lines.
 318                          *
 319                          * Note that uuencode/uudecode uses only the portable
 320                          * character set for encoded data and the portable
 321                          * character set characters must be represented in
 322                          * a single byte.  We use this knowledge to reuse
 323                          * buffer space while decoding.
 324                          */
 325                         octets = n;
 326                         obp = (unsigned char *) &inbuf[0];
 327                         ibp = &inbuf[1];
 328                         while (octets > 0) {
 329                                 if ((ch = outdec((unsigned char *)obp,
 330                                     (unsigned char *)ibp, octets))
 331                                     != 0x20) {
 332                                         /* invalid characters where detected */
 333                                         if (!warned) {
 334                                                 warned = 1;
 335                                                 (void) fprintf(stderr,
 336                                                     gettext("Invalid character"
 337                                                         " (0x%x) on line"
 338                                                         " %lld\n"), ch, line);
 339                                         }
```



```
 340                                         break;
 341                                 }
 342                                 ibp += 4;
 343                                 obp += 3;
 344                                 octets -= 3;
 345                         }
 346                         /*
 347                          * Only write out uncorrupted lines
 348                          */
 349                         if (octets <= 0) {
 350                                 (void) fwrite(inbuf, n, 1, out);
 351                         }
 352                 }
 353         } else {        /* use base64 algorithm */
 354                 endseen = numbase64chars = 0;
 355                 optr = outbuf;
 356                 while ((fgets(inbuf, sizeof (inbuf), in)) != NULL) {
 357                         /* process an input line */
 358                         iptr = inbuf;
 359                         while ((curchr = *(iptr++)) != '\0') {
 359                         while ((curchr = *(iptr++)) != NULL) {
 360                                 /* decode chars */
 361                                 if (curchr == '=') /* if end */
 362                                         endseen++;

 364                                 if (validbase64(curchr))
 365                                         chr[numbase64chars++] = curchr;
 366                                 /*
 367                                  * if we've gathered 4 base64 octets
 368                                  * we need to decode and output them
 369                                  */
 370                                 if (numbase64chars == 4) {
 371                                         /*LINTED*/
 372                                         if (optr - outbuf > BUFSIZE - 3) {
 373                                                 (void) fwrite(outbuf,
 374                                                     /*LINTED*/
 375                                                     (size_t)(optr - outbuf),
 376                                                     1, out);
 377                                                 if (ferror(out)) {
 378                                                         perror(prog);
 379                                                         exit(6);
 380                                                 }
 381                                                 optr = outbuf;
 382                                         }
 383                                         octets = outdec64(optr, chr, 4);
 384                                         optr += octets;
 385                                         numbase64chars = 0;
 386                                 }
 387                         }
 388                         /*
 389                          * handle any remaining base64 octets at end
 390                          */
 391                         if (endseen && numbase64chars > 0) {
 392                                 octets = outdec64(optr, chr, numbase64chars);
 393                                 optr += octets;
 394                                 numbase64chars = 0;
 395                         }
 396                 }
 397                 /*
 398                  * if we have generated any additional output
 399                  * in the buffer, write it out
 400                  */
 401                 if (optr != outbuf) {
 402                         /*LINTED*/
 403                         (void) fwrite(outbuf, (size_t)(optr - outbuf),
 404                             1, out);
```

```
405                         if (ferror(out)) {
406                                 perror(prog);
407                                 exit(6);
408                         }
409                 }

411                 if (endseen == 0) {
412                         (void) fprintf(stderr, gettext("No end line\n"));
413                         exit(5);
414                 }
415         }
416 }
```

*unchanged_portion_omitted_*

```
**********************************************************
   36951 Sun Dec 14 23:31:15 2014
new/usr/src/cmd/chmod/chmod.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 517 /*
 518  *  parseargs - generate getopt-friendly argument list for backwards
 519  *              compatibility with earlier Solaris usage (eg, chmod -w
 520  *              foo).
 521  *
 522  *  assumes the existence of a static set of alternates to argc and argv,
 523  *  (namely, mac, and mav[]).
 524  *
 525  */

 527 static void
 528 parseargs(int ac, char *av[])
 529 {
 530         int i;                  /* current argument                     */
 531         int fflag;              /* arg list contains "--"               */
 532         size_t mav_num;         /* number of entries in mav[]           */

 534         /*
 535          * We add an extra argument slot, in case we need to jam a "--"
 536          * argument into the list.
 537          */

 539         mav_num = (size_t)ac+2;

 541         if ((mav = calloc(mav_num, sizeof (char *))) == NULL) {
 542                 perror("chmod");
 543                 exit(2);
 544         }

 546         /* scan for the use of "--" in the argument list */

 548         for (fflag = i = 0; i < ac; i ++) {
 549                 if (strcmp(av[i], "--") == 0)
 550                         fflag = 1;
 551         }

 553         /* process the arguments */

 555         for (i = mac = 0;
 556             (av[i] != NULL) && (av[i][0] != '\0');
 556             (av[i] != (char *)NULL) && (av[i][0] != (char)NULL);
 557             i++) {
 558                 if (!fflag && av[i][0] == '-') {
 559                         /*
 560                          *  If there is not already a "--" argument specified,
 561                          *  and the argument starts with '-' but does not
 562                          *  contain any of the official option letters, then it
 563                          *  is probably a mode argument beginning with '-'.
 564                          *  Force a "--" into the argument stream in front of
 565                          *  it.
 566                          */

 568                         if ((strchr(av[i], 'R') == NULL &&
 569                             strchr(av[i], 'f') == NULL) &&
 570                             strchr(av[i], '@') == NULL) {
 571                                 if ((mav[mac++] = strdup("--")) == NULL) {
 572                                         perror("chmod");
```



```
 573                                         exit(2);
 574                                 }
 575                         }
 576                 }

 578                 if ((mav[mac++] = strdup(av[i])) == NULL) {
 579                         perror("chmod");
 580                         exit(2);
 581                 }
 582         }

 584         mav[mac] = (char *)NULL;
 585 }
______unchanged_portion_omitted_
```

```
**********************************************************
   17846 Sun Dec 14 23:31:15 2014
new/usr/src/cmd/cmd-crypto/tpmadm/admin_cmds.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 642 /*ARGSUSED*/
 643 int
 644 cmd_init(TSS_HCONTEXT hContext, TSS_HTPM hTPM, int argc, char *argv[])
 645 {
 646         TSS_RESULT ret;
 647         TSS_HOBJECT hKeySRK;

 649         if (set_object_policy(hTPM, TSS_SECRET_MODE_POPUP,
 650             gettext("= TPM owner passphrase ="), 0, NULL))
 651                 return (ERR_FAIL);

 653         ret = Tspi_Context_CreateObject(hContext, TSS_OBJECT_TYPE_RSAKEY,
 654             TSS_KEY_TSP_SRK | TSS_KEY_AUTHORIZATION, &hKeySRK);
 655         if (ret) {
 656                 print_error(ret, gettext("Create storage root key"));
 657                 return (ERR_FAIL);
 658         }

 660         if (set_object_policy(hKeySRK, TSS_SECRET_MODE_SHA1, NULL,
 661             sizeof (well_known), well_known))
 662                 return (ERR_FAIL);

 664         ret = Tspi_TPM_TakeOwnership(hTPM, hKeySRK, (TSS_HKEY)NULL);
 664         ret = Tspi_TPM_TakeOwnership(hTPM, hKeySRK, NULL);
 665         if (ret == TPM_E_NO_ENDORSEMENT) {
 666                 if (createek(hContext, hTPM))
 667                         return (ERR_FAIL);
 668                 ret = Tspi_TPM_TakeOwnership(hTPM, hKeySRK, (TSS_HKEY)NULL);
 668                 ret = Tspi_TPM_TakeOwnership(hTPM, hKeySRK, NULL);
 669         }
 670         if (ret) {
 671                 print_error(ret, gettext("Take ownership"));
 672                 return (ERR_FAIL);
 673         }

 675         return (0);
 676 }

 678 /*
 679  * Auth
 680  */

 682 /*ARGSUSED*/
 683 int
 684 cmd_auth(TSS_HCONTEXT hContext, TSS_HTPM hTPM, int argc, char *argv[])
 685 {
 686         TSS_RESULT ret;
 687         TSS_HPOLICY hNewPolicy;

 689         if (set_object_policy(hTPM, TSS_SECRET_MODE_POPUP,
 690             gettext("= TPM owner passphrase ="), 0, NULL))
 691                 return (ERR_FAIL);

 693         /* policy object for new passphrase */
 694         ret = Tspi_Context_CreateObject(hContext, TSS_OBJECT_TYPE_POLICY,
 695             TSS_POLICY_USAGE, &hNewPolicy);
 696         if (ret) {
```



```
 697                 print_error(ret, gettext("Create policy object"));
 698                 return (ERR_FAIL);
 699         }
 700         if (set_policy_options(hNewPolicy, TSS_SECRET_MODE_POPUP,
 701             gettext("= New TPM owner passphrase ="), 0, NULL))
 702                 return (ERR_FAIL);

 704         ret = Tspi_ChangeAuth(hTPM, (TSS_HOBJECT)NULL, hNewPolicy);
 704         ret = Tspi_ChangeAuth(hTPM, NULL, hNewPolicy);
 705         if (ret && ret != TSP_ERROR(TSS_E_POLICY_NO_SECRET)) {
 706                 print_error(ret, gettext("Change authorization"));
 707                 return (ERR_FAIL);
 708         }

 710         return (0);
 711 }
______unchanged_portion_omitted_
```

```
**********************************************************
   90915 Sun Dec 14 23:31:15 2014
new/usr/src/cmd/cmd-inet/usr.sbin/ipsecutils/ikeadm.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 703 /*
 704  * The possible ident-type keywords that might be used on the command
 705  * line.  This is a superset of the ones supported by ipseckey, those
 706  * in the ike config file, and those in ike.preshared.
 707  */
 708 static keywdtab_t       idtypes[] = {
 709         /* ip, ipv4, and ipv6 are valid for preshared keys... */
 710         {SADB_IDENTTYPE_RESERVED,       "ip"},
 711         {SADB_IDENTTYPE_RESERVED,       "ipv4"},
 712         {SADB_IDENTTYPE_RESERVED,       "ipv6"},
 713         {SADB_IDENTTYPE_PREFIX,         "prefix"},
 714         {SADB_IDENTTYPE_PREFIX,         "ipv4-prefix"},
 715         {SADB_IDENTTYPE_PREFIX,         "ipv6-prefix"},
 716         {SADB_IDENTTYPE_PREFIX,         "subnet"},
 717         {SADB_IDENTTYPE_PREFIX,         "subnetv4"},
 718         {SADB_IDENTTYPE_PREFIX,         "subnetv6"},
 719         {SADB_IDENTTYPE_FQDN,           "fqdn"},
 720         {SADB_IDENTTYPE_FQDN,           "dns"},
 721         {SADB_IDENTTYPE_FQDN,           "domain"},
 722         {SADB_IDENTTYPE_FQDN,           "domainname"},
 723         {SADB_IDENTTYPE_USER_FQDN,      "user_fqdn"},
 724         {SADB_IDENTTYPE_USER_FQDN,      "mbox"},
 725         {SADB_IDENTTYPE_USER_FQDN,      "mailbox"},
 726         {SADB_X_IDENTTYPE_DN,           "dn"},
 727         {SADB_X_IDENTTYPE_DN,           "asn1dn"},
 728         {SADB_X_IDENTTYPE_GN,           "gn"},
 729         {SADB_X_IDENTTYPE_GN,           "asn1gn"},
 730         {SADB_X_IDENTTYPE_ADDR_RANGE,   "ipv4-range"},
 731         {SADB_X_IDENTTYPE_ADDR_RANGE,   "ipv6-range"},
 732         {SADB_X_IDENTTYPE_ADDR_RANGE,   "rangev4"},
 733         {SADB_X_IDENTTYPE_ADDR_RANGE,   "rangev6"},
 734         {SADB_X_IDENTTYPE_KEY_ID,       "keyid"},
 735         {0, NULL}
 735         {NULL,  0}
 736 };
______unchanged_portion_omitted_

 883 /*
 884  * Preshared key field types...used for parsing preshared keys that
 885  * have been entered on the command line.  The code to parse preshared
 886  * keys (parse_ps, parse_key, parse_psfldid, parse_ikmtype, ...) is
 887  * mostly duplicated from in.iked's readps.c.
 888  */
 889 #define PSFLD_LOCID     1
 890 #define PSFLD_LOCIDTYPE 2
 891 #define PSFLD_REMID     3
 892 #define PSFLD_REMIDTYPE 4
 893 #define PSFLD_MODE      5
 894 #define PSFLD_KEY       6

 896 static keywdtab_t       psfldtypes[] = {
 897         {PSFLD_LOCID,           "localid"},
 898         {PSFLD_LOCIDTYPE,       "localidtype"},
 899         {PSFLD_REMID,           "remoteid"},
 900         {PSFLD_REMIDTYPE,       "remoteidtype"},
 901         {PSFLD_MODE,            "ike_mode"},
 902         {PSFLD_KEY,             "key"},
```



```
 903         {0, NULL}
 903         {NULL,  0}
 904 };
______unchanged_portion_omitted_

 925 static keywdtab_t       ikemodes[] = {
 926         {IKE_XCHG_IDENTITY_PROTECT,     "main"},
 927         {IKE_XCHG_AGGRESSIVE,           "aggressive"},
 928         {IKE_XCHG_IP_AND_AGGR,          "both"},
 929         {0, NULL}
 929         {NULL,  0}
 930 };
______unchanged_portion_omitted_

3279 #define REQ_ARG_CNT     1

3281 /*ARGSUSED*/
3282 static void
3283 parseit(int argc, char **argv, char *notused, boolean_t notused_either)
3284 {
3285         int     cmd, cmd_obj_args = 1;
3286         char    *cmdstr, *objstr;

3288         if (interactive) {
3289                 if (argc == 0)
3290                         return;
3291         }

3293         if (argc < REQ_ARG_CNT) {
3294                 usage();
3295         }

3297         cmdstr = argv[0];
3298         if (argc > REQ_ARG_CNT) {
3299                 cmd_obj_args++;
3300                 objstr = argv[1];
3301         } else {
3302                 objstr = NULL;
3303         }
3304         cmd = parsecmd(cmdstr, objstr);

3306         /* skip over args specifying command/object */
3307         argc -= cmd_obj_args;
3308         argv += cmd_obj_args;

3310         switch (cmd) {
3311         case IKE_SVC_GET_DEFS:
3312                 if (argc != 0) {
3313                         print_get_help();
3314                         break;
3315                 }
3316                 do_getdefs(cmd);
3317                 break;
3318         case IKE_SVC_GET_DBG:
3319         case IKE_SVC_GET_PRIV:
3320                 if (argc != 0) {
3321                         print_get_help();
3322                         break;
3323                 }
3324                 do_getvar(cmd);
3325                 break;
3326         case IKE_SVC_GET_STATS:
3327                 if (argc != 0) {
3328                         print_get_help();
3329                         break;
3330                 }
```

```
3331                do_getstats(cmd);
3332                break;
3333        case IKE_SVC_SET_DBG:
3334        case IKE_SVC_SET_PRIV:
3335                do_setvar(cmd, argc, argv);
3336                break;
3337        case IKE_SVC_SET_PIN:
3338        case IKE_SVC_DEL_PIN:
3339                do_setdel_pin(cmd, argc, argv);
3340                break;
3341        case IKE_SVC_DUMP_P1S:
3342        case IKE_SVC_DUMP_RULES:
3343        case IKE_SVC_DUMP_GROUPS:
3344        case IKE_SVC_DUMP_ENCRALGS:
3345        case IKE_SVC_DUMP_AUTHALGS:
3346        case IKE_SVC_DUMP_PS:
3347        case IKE_SVC_DUMP_CERTCACHE:
3348                if (argc != 0) {
3348                if (argc != NULL) {
3349                        print_dump_help();
3350                        break;
3351                }
3352                do_dump(cmd);
3353                break;
3354        case IKE_SVC_GET_P1:
3355        case IKE_SVC_GET_RULE:
3356        case IKE_SVC_GET_PS:
3357        case IKE_SVC_DEL_P1:
3358        case IKE_SVC_DEL_RULE:
3359        case IKE_SVC_DEL_PS:
3360                do_getdel(cmd, argc, argv);
3361                break;
3362        case IKE_SVC_NEW_RULE:
3363        case IKE_SVC_NEW_PS:
3364                do_new(cmd, argc, argv);
3365                break;
3366        case IKE_SVC_FLUSH_P1S:
3367        case IKE_SVC_FLUSH_CERTCACHE:
3368                if (argc != 0) {
3369                        print_flush_help();
3370                        break;
3371                }
3372                do_flush(cmd);
3373                break;
3374        case IKE_SVC_READ_RULES:
3375        case IKE_SVC_READ_PS:
3376        case IKE_SVC_WRITE_RULES:
3377        case IKE_SVC_WRITE_PS:
3378                do_rw(cmd, argc, argv);
3379                break;
3380        case IKEADM_HELP_GENERAL:
3381                print_help();
3382                break;
3383        case IKEADM_HELP_GET:
3384                print_get_help();
3385                break;
3386        case IKEADM_HELP_SET:
3387                print_set_help();
3388                break;
3389        case IKEADM_HELP_ADD:
3390                print_add_help();
3391                break;
3392        case IKEADM_HELP_DEL:
3393                print_del_help();
3394                break;
3395        case IKEADM_HELP_DUMP:
```



```
3396                print_dump_help();
3397                break;
3398        case IKEADM_HELP_FLUSH:
3399                print_flush_help();
3400                break;
3401        case IKEADM_HELP_READ:
3402                print_read_help();
3403                break;
3404        case IKEADM_HELP_WRITE:
3405                print_write_help();
3406                break;
3407        case IKEADM_HELP_TOKEN:
3408                print_token_help();
3409                break;
3410        case IKEADM_HELP_HELP:
3411                print_help_help();
3412                break;
3413        case IKEADM_EXIT:
3414                if (interactive)
3415                        exit(0);
3416                break;
3417        case IKE_SVC_DBG_RBDUMP:
3418                do_rbdump();
3419                break;
3420        case IKE_SVC_ERROR:
3421                usage();
3422        default:
3423                exit(0);
3424        }
3425 }
```

*unchanged_portion_omitted_*

```
**********************************************************
   28380 Sun Dec 14 23:31:16 2014
new/usr/src/cmd/fm/modules/common/fabric-xlate/fx_fabric.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
```

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

/*
 * Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
#include <stddef.h>
#include <strings.h>
#include <sys/fm/util.h>

#include "fabric-xlate.h"

#define FAB_LOOKUP(sz, name, field) \
        (void) nvlist_lookup_uint ## sz(nvl, name, field)

static boolean_t fab_xlate_fake_rp = B_TRUE;
static fab_err_tbl_t *fab_master_err_tbl;

/*
 * Translation tables for converting "fabric" error bits into "pci" ereports.
 * <Ereport Class Name>, <Error Bit Mask>, <Preparation Function>
 */

/* MACRO for table entries with no TGT ereports */
#define NT(class, bit, prep) class, bit, prep, NULL
/* Translate Fabric ereports to ereport.io.pci.* */
fab_erpt_tbl_t fab_pci_erpt_tbl[] = {
        PCI_DET_PERR,           PCI_STAT_PERROR,        NULL,
        PCI_MDPE,               PCI_STAT_S_PERROR,      NULL,
        PCI_SIG_SERR,           PCI_STAT_S_SYSERR,      NULL,
        PCI_MA,                 PCI_STAT_R_MAST_AB,     NULL,
        PCI_REC_TA,             PCI_STAT_R_TARG_AB,     NULL,
        PCI_SIG_TA,             PCI_STAT_S_TARG_AB,     NULL,
        NULL, 0 , NULL
        NULL, NULL, NULL
};

/* Translate Fabric ereports to ereport.io.pci.sec-* */
static fab_erpt_tbl_t fab_pci_bdg_erpt_tbl[] = {
        PCI_DET_PERR,           PCI_STAT_PERROR,        NULL,
```



```
        PCI_MDPE,               PCI_STAT_S_PERROR,      NULL,
        PCI_REC_SERR,           PCI_STAT_S_SYSERR,      NULL,
#ifdef sparc
        PCI_MA,                 PCI_STAT_R_MAST_AB,     NULL,
#endif
        PCI_REC_TA,             PCI_STAT_R_TARG_AB,     NULL,
        PCI_SIG_TA,             PCI_STAT_S_TARG_AB,     NULL,
        NULL, 0 , NULL
        NULL, NULL, NULL, NULL,
};


/* Translate Fabric ereports to ereport.io.pci.dto */
static fab_erpt_tbl_t fab_pci_bdg_ctl_erpt_tbl[] = {
        PCI_DTO,        PCI_BCNF_BCNTRL_DTO_STAT,       NULL,
        NULL, 0, NULL
        NULL, NULL, NULL
};

/* Translate Fabric ereports to ereport.io.pciex.* */
static fab_erpt_tbl_t fab_pcie_ce_erpt_tbl[] = {
        PCIEX_RE,       PCIE_AER_CE_RECEIVER_ERR,       NULL,
        PCIEX_RNR,      PCIE_AER_CE_REPLAY_ROLLOVER,    NULL,
        PCIEX_RTO,      PCIE_AER_CE_REPLAY_TO,          NULL,
        PCIEX_BDP,      PCIE_AER_CE_BAD_DLLP,           NULL,
        PCIEX_BTP,      PCIE_AER_CE_BAD_TLP,            NULL,
        PCIEX_ANFE,     PCIE_AER_CE_AD_NFE,             NULL,
        NULL, 0, NULL
        NULL, NULL, NULL
};

/*
 * Translate Fabric ereports to ereport.io.pciex.*
 * The Target Ereports for this section is only used on leaf devices, with the
 * exception of TO
 */
static fab_erpt_tbl_t fab_pcie_ue_erpt_tbl[] = {
        PCIEX_TE,       PCIE_AER_UCE_TRAINING,          NULL,
        PCIEX_DLP,      PCIE_AER_UCE_DLP,               NULL,
        PCIEX_SD,       PCIE_AER_UCE_SD,                NULL,
        PCIEX_ROF,      PCIE_AER_UCE_RO,                NULL,
        PCIEX_FCP,      PCIE_AER_UCE_FCP,               NULL,
        PCIEX_MFP,      PCIE_AER_UCE_MTLP,              NULL,
        PCIEX_CTO,      PCIE_AER_UCE_TO,                PCI_TARG_MA,
        PCIEX_UC,       PCIE_AER_UCE_UC,                NULL,
        PCIEX_ECRC,     PCIE_AER_UCE_ECRC,              NULL,
        PCIEX_CA,       PCIE_AER_UCE_CA,                PCI_TARG_REC_TA,
#ifdef sparc
        PCIEX_UR,       PCIE_AER_UCE_UR,                PCI_TARG_MA,
#endif
        PCIEX_POIS,     PCIE_AER_UCE_PTLP,              PCI_TARG_MDPE,
        NULL, 0, NULL
        NULL, NULL, NULL
};

/* Translate Fabric ereports to ereport.io.pciex.* */
static fab_erpt_tbl_t fab_pcie_sue_erpt_tbl[] = {
        PCIEX_S_TA_SC,  PCIE_AER_SUCE_TA_ON_SC,         PCI_TARG_REC_TA,
        PCIEX_S_MA_SC,  PCIE_AER_SUCE_MA_ON_SC,         PCI_TARG_MA,
        PCIEX_S_RTA,    PCIE_AER_SUCE_RCVD_TA,          PCI_TARG_REC_TA,
#ifdef sparc
        PCIEX_S_RMA,    PCIE_AER_SUCE_RCVD_MA,          PCI_TARG_MA,
#endif
        PCIEX_S_USC,    PCIE_AER_SUCE_USC_ERR,          NULL,
        PCIEX_S_USCMD,  PCIE_AER_SUCE_USC_MSG_DATA_ERR, PCI_TARG_REC_TA,
        PCIEX_S_UDE,    PCIE_AER_SUCE_UC_DATA_ERR,      PCI_TARG_MDPE,
```

```
121         PCIEX_S_UAT,    PCIE_AER_SUCE_UC_ATTR_ERR,      PCI_TARG_MDPE,
122         PCIEX_S_UADR,   PCIE_AER_SUCE_UC_ADDR_ERR,      PCI_TARG_MDPE,
123         PCIEX_S_TEX,    PCIE_AER_SUCE_TIMER_EXPIRED,    NULL,
124         PCIEX_S_PERR,   PCIE_AER_SUCE_PERR_ASSERT,      PCI_TARG_MDPE,
125         PCIEX_S_SERR,   PCIE_AER_SUCE_SERR_ASSERT,      NULL,
126         PCIEX_INTERR,   PCIE_AER_SUCE_INTERNAL_ERR,     NULL,
127         NULL, 0, NULL
127         NULL, NULL, NULL
128 };

130 /* Translate Fabric ereports to ereport.io.pcix.* */
131 static fab_erpt_tbl_t fab_pcix_erpt_tbl[] = {
132         PCIX_SPL_DIS,           PCI_PCIX_SPL_DSCD,      NULL,
133         PCIX_UNEX_SPL,          PCI_PCIX_UNEX_SPL,      NULL,
134         PCIX_RX_SPL_MSG,        PCI_PCIX_RX_SPL_MSG,    NULL,
135         NULL, 0, NULL
135         NULL, NULL, NULL
136 };
137 static fab_erpt_tbl_t *fab_pcix_bdg_erpt_tbl = fab_pcix_erpt_tbl;

139 /* Translate Fabric ereports to ereport.io.pcix.sec-* */
140 static fab_erpt_tbl_t fab_pcix_bdg_sec_erpt_tbl[] = {
141         PCIX_SPL_DIS,           PCI_PCIX_BSS_SPL_DSCD,  NULL,
142         PCIX_UNEX_SPL,          PCI_PCIX_BSS_UNEX_SPL,  NULL,
143         PCIX_BSS_SPL_OR,        PCI_PCIX_BSS_SPL_OR,    NULL,
144         PCIX_BSS_SPL_DLY,       PCI_PCIX_BSS_SPL_DLY,   NULL,
145         NULL, 0, NULL
145         NULL, NULL, NULL
146 };

148 /* Translate Fabric ereports to ereport.io.pciex.* */
149 static fab_erpt_tbl_t fab_pcie_nadv_erpt_tbl[] = {
150 #ifdef sparc
151         PCIEX_UR,               PCIE_DEVSTS_UR_DETECTED,        NULL,
152 #endif
153         PCIEX_FAT,              PCIE_DEVSTS_FE_DETECTED,        NULL,
154         PCIEX_NONFAT,           PCIE_DEVSTS_NFE_DETECTED,       NULL,
155         PCIEX_CORR,             PCIE_DEVSTS_CE_DETECTED,        NULL,
156         NULL, 0, NULL
156         NULL, NULL, NULL
157 };

159 /* Translate Fabric ereports to ereport.io.pciex.* */
160 static fab_erpt_tbl_t fab_pcie_rc_erpt_tbl[] = {
161         PCIEX_RC_FE_MSG,        PCIE_AER_RE_STS_FE_MSGS_RCVD,   NULL,
162         PCIEX_RC_NFE_MSG,       PCIE_AER_RE_STS_NFE_MSGS_RCVD,  NULL,
163         PCIEX_RC_CE_MSG,        PCIE_AER_RE_STS_CE_RCVD,        NULL,
164         PCIEX_RC_MCE_MSG,       PCIE_AER_RE_STS_MUL_CE_RCVD,    NULL,
165         PCIEX_RC_MUE_MSG,       PCIE_AER_RE_STS_MUL_FE_NFE_RCVD, NULL,
166         NULL, 0, NULL
166         NULL, NULL, NULL
167 };

169 /*
170  * Translate Fabric ereports to pseudo ereport.io.pciex.* RC Fabric Messages.
171  * If the RP is not a PCIe compliant RP or does not support AER, rely on the
172  * leaf fabric ereport to help create a xxx_MSG ereport coming from the RC.
173  */
174 static fab_erpt_tbl_t fab_pcie_fake_rc_erpt_tbl[] = {
175         PCIEX_RC_FE_MSG,        PCIE_DEVSTS_FE_DETECTED,        NULL,
176         PCIEX_RC_NFE_MSG,       PCIE_DEVSTS_NFE_DETECTED,       NULL,
177         PCIEX_RC_CE_MSG,        PCIE_DEVSTS_CE_DETECTED,        NULL,
178         NULL, 0, NULL
178         NULL, NULL, NULL,
179 };
```

*unchanged_portion_omitted_*

```
**********************************************************
   18915 Sun Dec 14 23:31:16 2014
new/usr/src/cmd/lp/cmd/lpsched/fncs.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 885 void GetRequestFiles(REQUEST *req, char *buffer, int length)
 886 {
 887         char buf[BUFSIZ];

 889         memset(buf, 0, sizeof(buf));

 891         if (req->title) {
 892                 char *r = req->title;
 893                 char *ptr = buf;

 895                 while ( *r && strncmp(r,"\\n",2)) {
 896                         *ptr++ = *r++;
 897                 }
 898         } else if (req->file_list)
 899                 strlcpy(buf, *req->file_list, sizeof (buf));
 900
 901         if (*buf == '\0' || !strncmp(buf, SPOOLDIR, sizeof(SPOOLDIR)-1))
 901         if (*buf == NULL || !strncmp(buf, SPOOLDIR, sizeof(SPOOLDIR)-1))
 902                 strcpy(buf, "<File name not available>");

 904         if (strlen(buf) > (size_t) 24) {
 905                 char *r;

 907                 if (r = strrchr(buf, '/'))
 908                         r++;
 909                 else
 910                         r = buf;
 911
 912                 snprintf(buffer, length, "%-.24s", r);
 913         } else
 914                 strlcpy(buffer, buf, length);
 915         return;
 916 }
______unchanged_portion_omitted_
```

****
97139 Sun Dec 14 23:31:17 2014
new/usr/src/cmd/lvm/rpc.mdcommd/mdmn_commd_server.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
****
_____***unchanged_portion_omitted_***

```
 202 #define FLUSH_DEBUGFILE() \
 203         if (commdout != (FILE *)NULL) { \
 204                 (void) fflush(commdout); \
 205                 (void) fsync(fileno(commdout)); \
 206         }

 208 static void
 209 panic_system(int nid, md_mn_msgtype_t type, int master_err, int master_exitval,
 210     md_mn_result_t *slave_result)
 211 {
 212         md_mn_commd_err_t       commd_err;
 213         md_error_t              mne = mdnullerror;
 214         char                    *msg_buf;

 216         msg_buf = (char *)calloc(MAXPATHLEN + 1, sizeof (char));

 218         FLUSH_DEBUGFILE();

 220         if (master_err != MDMNE_ACK) {
 221                 (void) snprintf(msg_buf, MAXPATHLEN, "rpc.mdcommd: RPC "
 222                     "fail on master when processing message type %d\n", type);
 223         } else if (slave_result == NULL) {
 224                 (void) snprintf(msg_buf, MAXPATHLEN, "rpc.mdcommd: RPC fail "
 225                     "on node %d when processing message type %d\n", nid, type);
 226         } else {
 227                 (void) snprintf(msg_buf, MAXPATHLEN, "rpc.mdcommd: "
 228                     "Inconsistent return value from node %d when processing "
 229                     "message type %d. Master exitval = %d, "
 230                     "Slave exitval = %d\n", nid, type, master_exitval,
 231                     slave_result->mmr_exitval);
 232         }
 233         commd_err.size = strlen(msg_buf);
 234         commd_err.md_message = (uint64_t)(uintptr_t)&msg_buf[0];

 236         (void) metaioctl(MD_MN_COMMD_ERR, &commd_err, &mne, "rpc.mdcommd");
 237         (void) uadmin(A_DUMP, AD_BOOT, (uintptr_t)NULL);
 237         (void) uadmin(A_DUMP, AD_BOOT, NULL);
 238 }
```

_____***unchanged_portion_omitted_***

****
55676 Sun Dec 14 23:31:17 2014
new/usr/src/cmd/mdb/common/mdb/mdb_termio.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
****
_____***unchanged_portion_omitted_***

```
 315 static termio_info_t termio_info;

 317 static const termio_attr_t termio_attrs[] = {
 318         { "cub1", TIO_ATTR_REQSTR, &termio_info.ti_cub1 },
 319         { "cuf1", TIO_ATTR_REQSTR, &termio_info.ti_cuf1 },
 320         { "cuu1", TIO_ATTR_REQSTR, &termio_info.ti_cuu1 },
 321         { "cud1", TIO_ATTR_REQSTR, &termio_info.ti_cud1 },
 322         { "pad", TIO_ATTR_STR, &termio_info.ti_pad },
 323         { "el", TIO_ATTR_REQSTR, &termio_info.ti_el },
 324         { "am", TIO_ATTR_BOOL, &termio_info.ti_am },
 325         { "bw", TIO_ATTR_BOOL, &termio_info.ti_bw },
 326         { "npc", TIO_ATTR_BOOL, &termio_info.ti_npc },
 327         { "xenl", TIO_ATTR_BOOL, &termio_info.ti_xenl },
 328         { "xon", TIO_ATTR_BOOL, &termio_info.ti_xon },
 329         { "cols", TIO_ATTR_INT, &termio_info.ti_cols },
 330         { "lines", TIO_ATTR_INT, &termio_info.ti_lines },
 331         { "pb", TIO_ATTR_INT, &termio_info.ti_pb },
 332         { "smso", TIO_ATTR_STR, &termio_info.ti_smso },
 333         { "rmso", TIO_ATTR_STR, &termio_info.ti_rmso },
 334         { "smul", TIO_ATTR_STR, &termio_info.ti_smul },
 335         { "rmul", TIO_ATTR_STR, &termio_info.ti_rmul },
 336         { "enacs", TIO_ATTR_STR, &termio_info.ti_enacs },
 337         { "smacs", TIO_ATTR_STR, &termio_info.ti_smacs },
 338         { "rmacs", TIO_ATTR_STR, &termio_info.ti_rmacs },
 339         { "smcup", TIO_ATTR_STR, &termio_info.ti_smcup },
 340         { "rmcup", TIO_ATTR_STR, &termio_info.ti_rmcup },
 341         { "rev", TIO_ATTR_STR, &termio_info.ti_rev },
 342         { "bold", TIO_ATTR_STR, &termio_info.ti_bold },
 343         { "dim", TIO_ATTR_STR, &termio_info.ti_dim },
 344         { "sgr0", TIO_ATTR_STR, &termio_info.ti_sgr0 },
 345         { "smir", TIO_ATTR_STR, &termio_info.ti_smir },
 346         { "rmir", TIO_ATTR_STR, &termio_info.ti_rmir },
 347         { "ich1", TIO_ATTR_STR, &termio_info.ti_ich1 },
 348         { "ip", TIO_ATTR_STR, &termio_info.ti_ip },
 349         { "clear", TIO_ATTR_STR, &termio_info.ti_clear },
 350         { "cnorm", TIO_ATTR_STR, &termio_info.ti_cnorm },
 351         { "nel", TIO_ATTR_STR, &termio_info.ti_nel },
 352         { "cr", TIO_ATTR_STR, &termio_info.ti_cr },
 353         { NULL, 0, NULL }
 353         { NULL, NULL, NULL }
 354 };
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   12730 Sun Dec 14 23:31:18 2014
new/usr/src/cmd/mdb/common/modules/genunix/sysevent.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 181 int
 182 sysevent_subclass_list_walk_init(mdb_walk_state_t *wsp)
 183 {
 184         if (wsp->walk_addr == (uintptr_t)NULL) {
 184         if (wsp->walk_addr == NULL) {
 185                 mdb_warn("sysevent_subclass_list does not support global "
 186                     "walks");
 187                 return (WALK_ERR);
 188         }

 190         wsp->walk_data = mdb_alloc(sizeof (subclass_lst_t), UM_SLEEP);
 191         return (WALK_NEXT);
 192 }

 194 int
 195 sysevent_subclass_list_walk_step(mdb_walk_state_t *wsp)
 196 {
 197         int status;

 199         if (wsp->walk_addr == (uintptr_t)NULL)
 199         if (wsp->walk_addr == NULL)
 200                 return (WALK_DONE);

 202         if (mdb_vread(wsp->walk_data, sizeof (subclass_lst_t),
 203             wsp->walk_addr) == -1) {
 204                 mdb_warn("failed to read class list at %p", wsp->walk_addr);
 205                 return (WALK_ERR);
 206         }

 208         status = wsp->walk_callback(wsp->walk_addr, wsp->walk_data,
 209             wsp->walk_cbdata);

 211         wsp->walk_addr =
 212             (uintptr_t)(((subclass_lst_t *)wsp->walk_data)->sl_next);

 214         return (status);
 215 }
______unchanged_portion_omitted_

 228 int
 229 sysevent_class_list_walk_init(mdb_walk_state_t *wsp)
 230 {
 231         class_walk_data_t *cl_walker;

 233         if (wsp->walk_addr == (uintptr_t)NULL) {
 233         if (wsp->walk_addr == NULL) {
 234                 mdb_warn("sysevent_class_list does not support global walks");
 235                 return (WALK_ERR);
 236         }

 238         cl_walker = mdb_zalloc(sizeof (class_walk_data_t), UM_SLEEP);
 239         if (mdb_vread(cl_walker->hash_tbl,
 240             sizeof (cl_walker->hash_tbl), wsp->walk_addr) == -1) {
 241                 mdb_warn("failed to read class hash table at %p",
 242                     wsp->walk_addr);
 243                 return (WALK_ERR);
 244         }
```



```
 246         wsp->walk_addr = (uintptr_t)cl_walker->hash_tbl[0];
 247         wsp->walk_data = cl_walker;

 249         return (WALK_NEXT);
 250 }

 252 int
 253 sysevent_class_list_walk_step(mdb_walk_state_t *wsp)
 254 {
 255         int status = WALK_NEXT;
 256         class_walk_data_t *cl_walker;
 257         class_lst_t clist;

 259         cl_walker = (class_walk_data_t *)wsp->walk_data;

 261         /* Skip over empty class table entries */
 262         if (wsp->walk_addr != (uintptr_t)NULL) {
 262         if (wsp->walk_addr != NULL) {
 263                 if (mdb_vread(&clist, sizeof (class_lst_t),
 264                     wsp->walk_addr) == -1) {
 265                         mdb_warn("failed to read class list at %p",
 266                             wsp->walk_addr);
 267                         return (WALK_ERR);
 268                 }

 270                 status = wsp->walk_callback(wsp->walk_addr, NULL,
 271                     wsp->walk_cbdata);
 272                 wsp->walk_addr = (uintptr_t)clist.cl_next;
 273         } else {
 274                 if (cl_walker->hash_index > CLASS_HASH_SZ) {
 275                         return (WALK_DONE);
 276                 } else {
 277                         wsp->walk_addr = (uintptr_t)
 278                             cl_walker->hash_tbl[cl_walker->hash_index];
 279                         cl_walker->hash_index++;
 280                 }
 281         }


 284         return (status);
 285 }
______unchanged_portion_omitted_
```

```
**********************************************************
   20304 Sun Dec 14 23:31:18 2014
new/usr/src/cmd/mdb/common/modules/usba/usb.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_


  70 /*
  71  * Callback for usba_device2dip.
  72  * Callback called from the devinfo_children walk invoked in usba_device2dip.
  73  *
  74  * For the current dip, get the (potential) pointer to its usba_device_t
  75  * struct.
  76  * See if this pointer matches the address of the usba_device_t we're looking
  77  * for (passed in as usb_dev_p).  If so, stuff its value in u2d_dip_addr,
  78  * and terminate the walk.
  79  *
  80  * - dip_addr is the address in core of the dip currently being processed by the
  81  * walk
  82  * - local_dip is a pointer to a copy of the struct dev_info in local memory
  83  * - cb_data is the addr of the callback arg the walker was invoked with
  84  * (passed through transparently from walk invoker).
  85  *
  86  * Returns:
  87  * - WALK_NEXT on success (match not found yet)
  88  * - WALK_ERR on errors.
  89  * - WALK_DONE is returned, cb_data.found is set to TRUE, and
  90  * *cb_data.u2d_dip_addr is set to the matched dip addr if a dip corresponding
  91  * to the desired usba_device_t* is found.
  92  */
  93 /*ARGSUSED*/
  94 static int
  95 find_dip(uintptr_t dip_addr, const void *local_dip, void *cb_arg)
  96 {
  97         uintptr_t                       cur_usb_dev;
  98         usba_device2devinfo_cbdata_t    *cb_data =
  99                                 (usba_device2devinfo_cbdata_t *)cb_arg;

 101         if ((cur_usb_dev = mdb_usba_get_usba_device(dip_addr)) == (uintptr_t)NUL
 101         if ((cur_usb_dev = mdb_usba_get_usba_device(dip_addr)) == NULL) {
 102                 /*
 103                  * If there's no corresponding usba_device_t, this dip isn't
 104                  * a usb node.  Might be an sd node.  Ignore it.
 105                  */

 107                 return (WALK_NEXT);
 108         }

 110         if (cur_usb_dev == cb_data->u2d_target_usb_dev_p) {
 111                 *cb_data->u2d_dip_addr = dip_addr;
 112                 cb_data->u2d_found = TRUE;

 114                 return (WALK_DONE);
 115         }

 117         return (WALK_NEXT);
 118 }
_____unchanged_portion_omitted_


 177 /*
 178  * Generic walker usba_list_entry_t walker.
 179  * Works for any usba_list_entry_t list.
```



```
 180  */
 181 int
 182 usba_list_walk_init(mdb_walk_state_t *wsp)
 183 {
 184         /* Must have a start addr.  */
 185         if (wsp->walk_addr == (uintptr_t)NULL) {
 185         if (wsp->walk_addr == NULL) {
 186                 mdb_warn("not a global walk.  Starting address required\n");

 188                 return (WALK_ERR);
 189         }

 191         return (WALK_NEXT);
 192 }


 195 /*
 196  * Generic list walker step routine.
 197  * NOTE: multiple walkers share this routine.
 198  */
 199 int
 200 usba_list_walk_step(mdb_walk_state_t *wsp)
 201 {
 202         int                     status;
 203         usba_list_entry_t       list_entry;

 205         if (mdb_vread(&list_entry, sizeof (usba_list_entry_t),
 206             (uintptr_t)wsp->walk_addr) == -1) {
 207                 mdb_warn("failed to read usba_list_entry_t at %p",
 208                     wsp->walk_addr);

 210                 return (WALK_ERR);
 211         }

 213         status = wsp->walk_callback(wsp->walk_addr, &list_entry,
 214             wsp->walk_cbdata);
 215         wsp->walk_addr = (uintptr_t)list_entry.next;

 217         /* Check if we're at the last element */
 218         if (wsp->walk_addr == (uintptr_t)NULL) {
 218         if (wsp->walk_addr == NULL) {

 220                 return (WALK_DONE);
 221         }

 223         return (status);
 224 }


 227 /*
 228  * usb_pipe_handle walker
 229  * Given a pointer to a usba_device_t, walk the array of endpoint
 230  * pipe_handle lists.
 231  * For each list, traverse the list, invoking the callback on each element.
 232  *
 233  * Note this function takes the address of a usba_device struct (which is
 234  * easily obtainable), but actually traverses a sub-portion of the struct
 235  * (which address is not so easily obtainable).
 236  */
 237 int
 238 usb_pipe_handle_walk_init(mdb_walk_state_t *wsp)
 239 {
 240         if (wsp->walk_addr == (uintptr_t)NULL) {
 240         if (wsp->walk_addr == NULL) {
 241                 mdb_warn("not a global walk; usba_device_t required\n");
```

```
243                 return (WALK_ERR);
244         }

246         wsp->walk_data = mdb_alloc((sizeof (usba_ph_impl_t)) * USBA_N_ENDPOINTS,
247                                    UM_SLEEP | UM_GC);

249         /*
250          * Read the usb_ph_list array into local memory.
251          * Set start address to first element/endpoint in usb_pipehandle_list
252          */
253         if (mdb_vread(wsp->walk_data,
254             (sizeof (usba_ph_impl_t)) * USBA_N_ENDPOINTS,
255             (uintptr_t)((size_t)(wsp->walk_addr) +
256             offsetof(usba_device_t, usb_ph_list))) == -1) {
257                 mdb_warn("failed to read usb_pipehandle_list at %p",
258                     wsp->walk_addr);

260                 return (WALK_ERR);
261         }

263         wsp->walk_arg = 0;

265         return (WALK_NEXT);
266 }
```

*unchanged_portion_omitted_*

```
394 /*
395  * usba_device walker:
396  *
397  * walks the chain of usba_device structs headed by usba_device_list in usba.c
398  * NOTE: It uses the generic list walk step routine usba_list_walk_step.
399  * No walk_fini routine is needed.
400  */
401 int
402 usba_device_walk_init(mdb_walk_state_t *wsp)
403 {
404         usba_list_entry_t       list_entry;

406         if (wsp->walk_addr != (uintptr_t)NULL) {
406         if (wsp->walk_addr != NULL) {
407                 mdb_warn(
408                     "global walk only.  Must be invoked without an address\n");

410                 return (WALK_ERR);
411         }

413         if (mdb_readvar(&list_entry, "usba_device_list") == -1) {
414                 mdb_warn("failed to read usba_device_list");

416                 return (WALK_ERR);
417         }

419         /* List head is not part of usba_device_t, get first usba_device_t */
420         wsp->walk_addr = (uintptr_t)list_entry.next;

422         return (WALK_NEXT);
423 }


426 /*
427  * usba_device dcmd
428  *      Given the address of a usba_device struct, dump summary info
429  *      -v:     Print more (verbose) info
430  *      -p:     Walk/dump all open pipes for this usba_device
431  */
```



```
432 /*ARGSUSED*/
433 int
434 usba_device(uintptr_t addr, uint_t flags, int argc, const mdb_arg_t *argv)
435 {
436         int             status;
437         char            pathname[MAXNAMELEN];
438         char            dname[MODMAXNAMELEN + 1] = "<unatt>"; /* Driver name */
439         char            drv_statep[MODMAXNAMELEN+ 10];
440         uint_t          usb_flag  = 0;
440         uint_t          usb_flag  = NULL;
441         boolean_t       no_driver_attached = FALSE;
442         uintptr_t       dip_addr;
443         struct dev_info devinfo;

445         if (!(flags & DCMD_ADDRSPEC)) {
446                 /* Global walk */
447                 if (mdb_walk_dcmd("usba_device", "usba_device", argc,
448                     argv) == -1) {
449                         mdb_warn("failed to walk usba_device");

451                         return (DCMD_ERR);
452                 }

454                 return (DCMD_OK);
455         }

457         if (mdb_getopts(argc, argv,
458             'p', MDB_OPT_SETBITS, USB_DUMP_ACTIVE_PIPES, &usb_flag,
459             'v', MDB_OPT_SETBITS, USB_DUMP_VERBOSE, &usb_flag, NULL) != argc) {

461                 return (DCMD_USAGE);
462         }

464         if (usb_flag && !(DCMD_HDRSPEC(flags))) {
465                 mdb_printf("\n");
466         }

468         if (DCMD_HDRSPEC(flags)) {
469                 mdb_printf("%<u>%-15s %4s %-?s %-42s%</u>\n",
470                     "NAME", "INST", "DIP", "PATH                                ");
471         }

473         status = usba_device2dip(addr, &dip_addr);
474         /*
475          * -1 = error
476          * 0 = no error, no match
477          * 1 = no error, match
478          */
479         if (status != 1) {
480                 if (status == -1) {
481                         mdb_warn("error looking for dip for usba_device %p",
482                             addr);
483                 } else {
484                         mdb_warn("failed to find dip for usba_device %p\n",
485                             addr);
486                 }
487                 mdb_warn("dip and statep unobtainable\n");

489                 return (DCMD_ERR);
490         }

492         /* Figure out what driver (name) is attached to this node. */
493         (void) mdb_devinfo2driver(dip_addr, (char *)dname, sizeof (dname));

495         if (mdb_vread(&devinfo, sizeof (struct dev_info),
496             dip_addr) == -1) {
```

```
497                 mdb_warn("failed to read devinfo");

499                 return (DCMD_ERR);
500         }

502         if (!(DDI_CF2(&devinfo))) {
503                 no_driver_attached = TRUE;
504         }

506         (void) mdb_ddi_pathname(dip_addr, pathname, sizeof (pathname));
507         mdb_printf("%-15s %2d   %-?p %s\n", dname, devinfo.devi_instance,
508             dip_addr, pathname);

510         if (usb_flag & USB_DUMP_VERBOSE) {
511                 int             i;
512                 uintptr_t       statep = (uintptr_t)NULL;
512                 uintptr_t       statep = NULL;
513                 char            *string_descr;
514                 char            **config_cloud, **conf_str_descr;
515                 usb_dev_descr_t usb_dev_descr;
516                 usba_device_t   usba_device_struct;

518                 if (mdb_vread(&usba_device_struct,
519                     sizeof (usba_device_t), addr) == -1) {
520                         mdb_warn("failed to read usba_device struct");

522                         return (DCMD_ERR);
523                 }

525                 mdb_printf("    usba_device: %-16p\n\n", (usba_device_t *)addr);

527                 if (mdb_vread(&usb_dev_descr, sizeof (usb_dev_descr),
528                     (uintptr_t)usba_device_struct.usb_dev_descr) == -1) {
529                         mdb_warn("failed to read usb_dev_descr_t struct");

531                         return (DCMD_ERR);
532                 }

534                 mdb_printf("\n    idVendor: 0x%04x idProduct: 0x%04x "
535                     "usb_addr: 0x%02x\n", usb_dev_descr.idVendor,
536                     usb_dev_descr.idProduct, usba_device_struct.usb_addr);

538                 /* Get the string descriptor string into local space. */
539                 string_descr = (char *)mdb_alloc(USB_MAXSTRINGLEN, UM_GC);

541                 if (usba_device_struct.usb_mfg_str == NULL) {
542                         (void) strcpy(string_descr, "<No Manufacturer String>");
543                 } else {
544                         if (mdb_readstr(string_descr, USB_MAXSTRINGLEN,
545                             (uintptr_t)usba_device_struct.usb_mfg_str) == -1) {
546                                 mdb_warn("failed to read manufacturer "
547                                     "string descriptor");
548                                 (void) strcpy(string_descr, "???");
549                         }
550                 }
551                 mdb_printf("\n    Manufacturer String:\t%s\n", string_descr);

553                 if (usba_device_struct.usb_product_str == NULL) {
554                         (void) strcpy(string_descr, "<No Product String>");
555                 } else {
556                         if (mdb_readstr(string_descr, USB_MAXSTRINGLEN,
557                             (uintptr_t)usba_device_struct.usb_product_str) ==
558                             -1) {
559                                 mdb_warn("failed to read product string "
560                                     "descriptor");
561                                 (void) strcpy(string_descr, "???");
```



```
562                         }
563                 }
564                 mdb_printf("    Product String:\t\t%s\n", string_descr);

566                 if (usba_device_struct.usb_serialno_str == NULL) {
567                         (void) strcpy(string_descr, "<No SerialNumber String>");
568                 } else {
569                         if (mdb_readstr(string_descr, USB_MAXSTRINGLEN,
570                             (uintptr_t)usba_device_struct.usb_serialno_str) ==
571                             -1) {
572                                 mdb_warn("failed to read serial number string "
573                                     "descriptor");
574                                 (void) strcpy(string_descr, "???");
575                         }
576                 }
577                 mdb_printf("    SerialNumber String:\t%s\n", string_descr);

579                 if (no_driver_attached) {
580                         mdb_printf("\n");
581                 } else {
582                         mdb_printf("      state_p: ");

584                         /*
585                          * Given the dip, find the associated statep. The
586                          * convention to generate this soft state anchor is:
587                          *      <driver_name>_statep
588                          */
589                         (void) mdb_snprintf(drv_statep, sizeof (drv_statep),
590                             "%s_statep", dname);
591                         if (mdb_devinfo2statep(dip_addr, drv_statep,
592                             &statep) == -1) {
593                                 mdb_warn("failed to find %s state struct for "
594                                     "dip %p", drv_statep, dip_addr);

596                                 return (DCMD_ERR);
597                         }
598                         mdb_printf("%-?p\n", statep);
599                 }

601                 config_cloud = (char **)mdb_alloc(sizeof (void *) *
602                     usba_device_struct.usb_n_cfgs, UM_GC);

604                 conf_str_descr = (char **)mdb_alloc(sizeof (void *) *
605                     usba_device_struct.usb_n_cfgs, UM_GC);

607                 if ((usba_device_struct.usb_cfg_array) &&
608                     (usba_device_struct.usb_cfg_str_descr)) {
609                         if ((mdb_vread(config_cloud,  sizeof (void *) *
610                             usba_device_struct.usb_n_cfgs,
611                             (uintptr_t)usba_device_struct.usb_cfg_array) ==
612                             -1) || (mdb_vread(conf_str_descr, sizeof (void *)
613                             * usba_device_struct.usb_n_cfgs, (uintptr_t)
614                             usba_device_struct.usb_cfg_str_descr)) == -1) {

616                                 mdb_warn("failed to read config cloud pointers");

618                         } else {

620                                 mdb_printf("\n    Device Config Clouds:\n"
621                                     "     Index\tConfig\t\tConfiguration "
622                                     "String\n"
623                                     "    -----\t------\t\t"
624                                     "--------------------\n");

626                                 for (i = 0; i < usba_device_struct.usb_n_cfgs;
627                                     i++) {
```

```
628                                     if (mdb_readstr(string_descr,
629                                         USB_MAXSTRINGLEN,
630                                         (uintptr_t)conf_str_descr[i]) ==
631                                         -1) {
632                                             (void) strcpy(string_descr,
633                                                 "<No Configuration "
634                                                 "String>");
635                                     }
636                                     mdb_printf("    %4d\t0x%p\t%s\n", i,
637                                         config_cloud[i], string_descr);
638                                 }
639                         }
640                 }

642                 mdb_printf("\n    Active configuration index: %d\n",
643                     usba_device_struct.usb_active_cfg_ndx);
644         }

646         if (usb_flag & USB_DUMP_ACTIVE_PIPES) {

648                 if (mdb_pwalk_dcmd("usb_pipe_handle", "usb_pipe_handle",
649                     0, NULL, addr) == -1) {
650                         mdb_warn("failed to walk usb_pipe_handle");
651                         return (DCMD_ERR);
652                 }
653         }

655         return (DCMD_OK);
656 }


659 /*
660  * Dump the contents of the usba_debug_buf, from the oldest to newest,
661  * wrapping around if necessary.
662  */
663 /*ARGSUSED*/
664 int
665 usba_debug_buf(uintptr_t addr, uint_t flags, int argc, const mdb_arg_t *argv)
666 {
667         char    *debug_buf_addr;        /* addr in core */
668         char    *local_debug_buf;       /* local copy of buf */
669         int     debug_buf_size;
670         char    *term_p;
671         int     being_cleared;

673         if (flags & DCMD_ADDRSPEC) {

675                 return (DCMD_USAGE);
676         }

678         if (mdb_readvar(&being_cleared, "usba_clear_debug_buf_flag") ==
679             -1) {
680                 mdb_warn("failed to read usba_clear_debug_buf_flag");

682                 return (DCMD_ERR);
683         }
684         if (being_cleared) {

686                 return (DCMD_OK);
687         }

689         if (mdb_readvar(&debug_buf_addr, "usba_debug_buf") == -1) {
690                 mdb_warn("failed to read usba_debug_buf");

692                 return (DCMD_ERR);
693         }
```



```
695         if (debug_buf_addr == NULL) {
696                 mdb_warn("usba_debug_buf not allocated\n");

698                 return (DCMD_OK);
699         }


702         if (mdb_readvar(&debug_buf_size, "usba_debug_buf_size") == -1) {
703                 mdb_warn("failed to read usba_debug_buf_size");

705                 return (DCMD_ERR);
706         }

708         debug_buf_size += USB_DEBUG_SIZE_EXTRA_ALLOC;
709         local_debug_buf = (char *)mdb_alloc(debug_buf_size, UM_SLEEP | UM_GC);

711         if ((mdb_vread(local_debug_buf, debug_buf_size,
712             (uintptr_t)debug_buf_addr)) == -1) {
713                 mdb_warn("failed to read usba_debug_buf at %p",
714                     local_debug_buf);

716                 return (DCMD_ERR);
717         }
718         local_debug_buf[debug_buf_size - 1] = '\0';

720         if (strlen(local_debug_buf) == 0) {
720         if (strlen(local_debug_buf) == NULL) {

722                 return (DCMD_OK);
723         }

725         if ((term_p = strstr(local_debug_buf, ">>>>")) == NULL) {
726                 mdb_warn("failed to find terminator \">>>>\"\n");

728                 return (DCMD_ERR);
729         }

731         /*
732          * Print the chunk of buffer from the terminator to the end.
733          * This will print a null string if no wrap has occurred yet.
734          */
735         mdb_printf("%s", term_p+5);     /* after >>>>\0 to end of buf */
736         mdb_printf("%s\n", local_debug_buf);    /* beg of buf to >>>>\0 */

738         return (DCMD_OK);
739 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    9782 Sun Dec 14 23:31:18 2014
new/usr/src/cmd/picl/plugins/common/piclevent/picl_slm.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 286 /*
 287  * piclslm_deliver_event - called by syseventd to deliver an event buffer.
 288  *                        The event buffer is subsequently delivered to
 289  *                        picld.  If picld, is not responding to the
 290  *                        delivery attempt, we will ignore it.
 291  */
 292 /*ARGSUSED*/
 293 static int
 294 piclslm_deliver_event(sysevent_t *ev, int flag)
 295 {
 296         sysevent_t      *dupev;
 297         nvlist_t        *nvl;
 298         char            *ec;
 299         char            *esc;
 300         char            *ename;
 301         int             retval;
 302         char            *pack_buf;
 303         size_t          nvl_size;
 304         int             rval;

 306         /*
 307          * Filter out uninteresting events
 308          */
 309         ec = sysevent_get_class_name(ev);
 310         esc = sysevent_get_subclass_name(ev);
 311         if (piclslm_debug)
 312                 syslog(LOG_INFO,
 313                     "picl_slm: got sysevent  ev:%p class:%s subclass:%s\n",
 314                     ev, (ec) ? ec : "NULL", (esc) ? esc : "NULL");
 315         if ((ec == NULL) || (esc == NULL)) {
 316                 return (0);
 317         } else if (strcmp(ec, EC_DEVFS) == 0) {
 318                 if (strcmp(esc, ESC_DEVFS_DEVI_ADD) == 0)
 319                         ename = strdup(PICLEVENT_SYSEVENT_DEVICE_ADDED);
 320                 else if (strcmp(esc, ESC_DEVFS_DEVI_REMOVE) == 0)
 321                         ename = strdup(PICLEVENT_SYSEVENT_DEVICE_REMOVED);
 322                 else
 323                         return (0);
 324         } else if (strcmp(ec, EC_DR) == 0) {
 325                 if (strcmp(esc, ESC_DR_AP_STATE_CHANGE) == 0)
 326                         ename = strdup(PICLEVENT_DR_AP_STATE_CHANGE);
 327                 else if (strcmp(esc, ESC_DR_REQ) == 0)
 328                         ename = strdup(PICLEVENT_DR_REQ);
 329                 else
 330                         return (0);
 331         } else {
 332                 return (0);
 333         }

 335         if (ename == NULL)
 336                 return (EAGAIN);

 338         /*
 339          * Make a copy to expand attribute list
 340          */
 341         dupev = sysevent_dup(ev);
 342         if (dupev == NULL) {
```



```
 343                 free(ename);
 344                 return (EAGAIN);
 345         }

 347         if (nvlist_alloc(&nvl, NV_UNIQUE_NAME_TYPE, 0)) {
 347         if (nvlist_alloc(&nvl, NV_UNIQUE_NAME_TYPE, NULL)) {
 348                 free(ename);
 349                 sysevent_free(dupev);
 350                 return (EAGAIN);
 351         }

 353         if (strcmp(ec, EC_DEVFS) == 0) {
 354                 rval = piclslm_add_ec_devfs_args(nvl, dupev);
 355         } else if (strcmp(ec, EC_DR) == 0) {
 356                 if (strcmp(esc, ESC_DR_REQ) == 0) {
 357                         rval = piclslm_add_ec_dr_req_args(nvl, dupev);
 358                 } else {
 359                         rval = piclslm_add_ec_dr_args(nvl, dupev);
 360                 }
 361         }

 363         if (rval != 0) {
 364                 free(ename);
 365                 nvlist_free(nvl);
 366                 sysevent_free(dupev);
 367                 return ((rval == EAGAIN) ? EAGAIN : 0);
 368         }

 370         pack_buf = NULL;
 371         if (nvlist_add_string(nvl, PICLEVENTARG_EVENT_NAME, ename) ||
 372             nvlist_add_string(nvl, PICLEVENTARG_DATA_TYPE,
 373             PICLEVENTARG_PICLEVENT_DATA) ||
 374             nvlist_pack(nvl, &pack_buf, &nvl_size, NV_ENCODE_NATIVE, 0)) {
 374             nvlist_pack(nvl, &pack_buf, &nvl_size, NV_ENCODE_NATIVE, NULL)) {
 375                 free(ename);
 376                 nvlist_free(nvl);
 377                 sysevent_free(dupev);
 378                 return (EAGAIN);
 379         }

 381         /*
 382          * Add nvlist_t to queue
 383          */
 384         (void) mutex_lock(&nvq_lock);
 385         retval = add_to_queue(pack_buf, nvl_size);
 386         (void) cond_signal(&nvq_cv);
 387         (void) mutex_unlock(&nvq_lock);

 389         nvlist_free(nvl);
 390         sysevent_free(dupev);
 391         free(ename);
 392         return (retval < 0 ? EAGAIN : 0);
 393 }

 395 struct slm_mod_ops *
 396 slm_init(void)
 397 {
 398         cleanup = 0;

 400         init_queue();

 402         (void) mutex_init(&nvq_lock, USYNC_THREAD, NULL);
 403         (void) cond_init(&nvq_cv, USYNC_THREAD, NULL);

 405         if (thr_create(NULL, 0, piclslm_deliver_thr,
 405         if (thr_create(NULL, NULL, piclslm_deliver_thr,
```

```
406             NULL, THR_BOUND, &piclslm_deliver_thr_id) != 0) {
407                 (void) mutex_destroy(&nvq_lock);
408                 (void) cond_destroy(&nvq_cv);
409                 return (NULL);
410         }
411         return (&piclslm_mod_ops);
412 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    9362 Sun Dec 14 23:31:19 2014
new/usr/src/cmd/powertop/common/powertop.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

  95 int
  96 main(int argc, char **argv)
  97 {
  98         double          interval, interval_usr;
  99         hrtime_t        interval_start;
 100         int             index2 = 0, c, dump_count = 0;
 101         char            *endptr, key;
 102         boolean_t       root_user = B_FALSE;
 103         struct pollfd   pollset;

 105         static struct option opts[] = {
 106                 { "dump", 1, NULL, 'd' },
 107                 { "time", 1, NULL, 't' },
 108                 { "help", 0, NULL, 'h' },
 109                 { "cpu", 1, NULL, 'c' },
 110                 { "verbose", 0, NULL, 'v' },
 111                 { 0, 0, NULL, 0 }
 112         };

 114         pt_set_progname(argv[0]);

 116         /*
 117          * Enumerate the system's CPUs, populate cpu_table, g_ncpus
 118          */
 119         if ((g_ncpus = g_ncpus_observed = pt_enumerate_cpus()) == 0)
 120                 exit(EXIT_FAILURE);

 122         if ((g_bit_depth = pt_get_bit_depth()) < 0)
 123                 exit(EXIT_FAILURE);

 125         g_features = 0;
 126         interval = interval_usr = INTERVAL_DEFAULT;
 127         g_op_mode = PT_MODE_DEFAULT;
 128         g_max_cstate = 0;
 129         g_argv = NULL;
 130         g_argc = 0;
 131         g_observed_cpu = 0;
 132         g_turbo_supported = B_FALSE;
 133         g_sig_resize = B_FALSE;
 134         g_curr_sugg = NULL;

 136         while ((c = getopt_long(argc, argv, "d:t:hvc:", opts, &index2))
 137             != EOF) {
 138                 if (c == -1)
 139                         break;

 141                 switch (c) {
 142                 case 'd':
 143                         if (PT_ON_DUMP) {
 144                                 pt_usage();
 145                                 exit(EXIT_USAGE);
 146                         }

 148                         g_op_mode |= PT_MODE_DUMP;
 149                         g_gui = B_FALSE;
 150                         dump_count = (int)strtod(optarg, &endptr);
```



```
 152                         if (dump_count <= 0 || *endptr != '\0') {
 152                         if (dump_count <= 0 || *endptr != NULL) {
 153                                 pt_usage();
 154                                 exit(EXIT_USAGE);
 155                         }

 157                         break;
 158                 case 't':
 159                         if (PT_ON_TIME) {
 160                                 pt_usage();
 161                                 exit(EXIT_USAGE);
 162                         }

 164                         g_op_mode |= PT_MODE_TIME;
 165                         interval = interval_usr = (double)strtod(optarg,
 166                             &endptr);

 168                         if (*endptr != '\0' || interval < 1 ||
 168                         if (*endptr != NULL || interval < 1 ||
 169                             interval > INTERVAL_MAX) {
 170                                 pt_usage();
 171                                 exit(EXIT_USAGE);
 172                         }

 174                         break;
 175                 case 'v':
 176                         if (PT_ON_CPU || PT_ON_VERBOSE) {
 177                                 pt_usage();
 178                                 exit(EXIT_USAGE);
 179                         }

 181                         g_op_mode |= PT_MODE_VERBOSE;
 182                         break;
 183                 case 'c':
 184                         if (PT_ON_CPU || PT_ON_VERBOSE) {
 185                                 pt_usage();
 186                                 exit(EXIT_USAGE);
 187                         }

 189                         g_op_mode |= PT_MODE_CPU;
 190                         g_observed_cpu = (uint_t)strtod(optarg, &endptr);

 192                         if (g_observed_cpu >= g_ncpus) {
 193                                 pt_usage();
 194                                 exit(EXIT_USAGE);
 195                         }

 197                         g_argc = 1;
 198                         g_ncpus_observed = 1;

 200                         if ((g_argv = malloc(sizeof (char *))) == NULL)
 201                                 return (EXIT_FAILURE);

 203                         if ((*g_argv = malloc(sizeof (char) * 5)) == NULL)
 204                                 return (EXIT_FAILURE);

 206                         (void) snprintf(*g_argv, 5, "%d\0", g_observed_cpu);
 207                         break;
 208                 case 'h':
 209                         pt_usage();
 210                         exit(EXIT_SUCCESS);
 211                 default:
 212                         pt_usage();
 213                         exit(EXIT_USAGE);
 214                 }
 215         }
```

```
217         if (optind < argc) {
218                 pt_usage();
219                 exit(EXIT_USAGE);
220         }

222         (void) printf("%s   %s\n\n", TITLE, COPYRIGHT_INTEL);

224         (void) printf("Collecting data for %.2f second(s) \n",
225             (float)interval);

227         /* Prepare P-state statistics */
228         if (pt_cpufreq_stat_prepare() == 0)
229                 g_features |= FEATURE_PSTATE;

231         /* Prepare C-state statistics */
232         if (pt_cpuidle_stat_prepare() == 0)
233                 g_features |= FEATURE_CSTATE;
234         else
235                 /*
236                  * PowerTop was unable to run a DTrace program,
237                  * most likely for lack of permissions.
238                  */
239                 exit(EXIT_FAILURE);

241         /* Prepare event statistics */
242         if (pt_events_stat_prepare() != -1)
243                 g_features |= FEATURE_EVENTS;

245         /*
246          * If the system is running on battery, find out what's
247          * the kstat module for it
248          */
249         pt_battery_mod_lookup();

251         /* Prepare turbo statistics */
252         if (pt_turbo_stat_prepare() == 0)
253                 g_features |= FEATURE_TURBO;

255         /*
256          * Initialize the display.
257          */
258         if (!PT_ON_DUMP) {
259                 pt_display_init_curses();
260                 pt_display_setup(B_FALSE);
261                 (void) signal(SIGWINCH, pt_sig_handler);

263                 pt_display_title_bar();
264                 pt_display_status_bar();

266                 g_gui = B_TRUE;
267                 pollset.fd = STDIN_FILENO;
268                 pollset.events = POLLIN;
269         }

271         /*
272          * Installs the initial suggestions, running as root and turning CPU
273          * power management ON.
274          */
275         if (geteuid() != 0) {
276                 pt_sugg_as_root();
277         } else {
278                 root_user = B_TRUE;
279                 pt_cpufreq_suggest();
280         }
```



```
282         while (true) {
283                 key = 0;

285                 if (g_sig_resize)
286                         pt_display_resize();

288                 interval_start = gethrtime();

290                 if (!PT_ON_DUMP) {
291                         if (poll(&pollset, (nfds_t)1,
292                             (int)(interval * MILLISEC)) > 0)
293                                 (void) read(STDIN_FILENO, &key, 1);
294                 } else {
295                         (void) sleep((int)interval);
296                 }

298                 g_interval_length = (double)(gethrtime() - interval_start)
299                     /NANOSEC;

301                 g_top_events = 0;
302                 g_total_events = 0;

304                 (void) memset(g_event_info, 0,
305                     EVENT_NUM_MAX * sizeof (event_info_t));
306                 (void) memset(g_cstate_info, 0,
307                     NSTATES * sizeof (state_info_t));

309                 /* Collect idle state transition stats */
310                 if (g_features & FEATURE_CSTATE &&
311                     pt_cpuidle_stat_collect(g_interval_length) < 0) {
312                         /* Reinitialize C-state statistics */
313                         if (pt_cpuidle_stat_prepare() != 0)
314                                 exit(EXIT_FAILURE);

316                         continue;
317                 }

319                 /* Collect frequency change stats */
320                 if (g_features & FEATURE_PSTATE &&
321                     pt_cpufreq_stat_collect(g_interval_length) < 0) {
322                         /* Reinitialize P-state statistics */
323                         if (pt_cpufreq_stat_prepare() != 0)
324                                 exit(EXIT_FAILURE);

326                         continue;
327                 }

329                 /* Collect event statistics */
330                 if (g_features & FEATURE_EVENTS &&
331                     pt_events_stat_collect() < 0) {
332                         /* Reinitialize event statistics */
333                         if (pt_events_stat_prepare() != 0)
334                                 exit(EXIT_FAILURE);

336                         continue;
337                 }

339                 /* Collect turbo statistics */
340                 if (g_features & FEATURE_TURBO &&
341                     pt_turbo_stat_collect() < 0)
342                         exit(EXIT_FAILURE);

344                 /* Show CPU power states */
345                 pt_display_states();

347                 /* Show wakeups events affecting PM */
```

```
348                 if (g_features & FEATURE_EVENTS) {
349                         pt_display_wakeups(g_interval_length);
350                         pt_display_events(g_interval_length);
351                 }

353                 pt_battery_print();

355                 if (key && !PT_ON_DUMP) {
356                         switch (toupper(key)) {
357                         case 'Q':
358                                 exit(EXIT_SUCCESS);
359                                 break;

361                         case 'R':
362                                 interval = 3;
363                                 break;
364                         }

366                         /*
367                          * Check if the user has activated the current
368                          * suggestion.
369                          */
370                         if (g_curr_sugg != NULL &&
371                             toupper(key) == g_curr_sugg->key &&
372                             g_curr_sugg->func)
373                                 g_curr_sugg->func();
374                 }

376                 if (dump_count)
377                         dump_count--;

379                 /* Exits if user requested a dump */
380                 if (PT_ON_DUMP && !dump_count)
381                         exit(EXIT_SUCCESS);

383                 /* No key pressed, will suggest something */
384                 if (!key && !dump_count)
385                         pt_sugg_pick();

387                 /* Refresh display */
388                 if (!PT_ON_DUMP)
389                         pt_display_update();

391                 if (root_user)
392                         pt_cpufreq_suggest();

394                 /*
395                  * Update the interval based on how long the CPU was in the
396                  * longest c-state during the last snapshot. If the user
397                  * specified an interval we skip this bit and keep it fixed.
398                  */
399                 if (g_features & FEATURE_CSTATE && !PT_ON_TIME &&
400                     g_longest_cstate > 0 &&
401                     g_cstate_info[g_longest_cstate].events > 0) {
402                         double deep_idle_res = (((double)
403                             g_cstate_info[g_longest_cstate].total_time/MICROSEC
404                             /g_ncpus)/g_cstate_info[g_longest_cstate].events);

406                         if (deep_idle_res < INTERVAL_DEFAULT ||
407                             (g_total_events/interval) < 1)
408                                 interval = INTERVAL_DEFAULT;
409                         else
410                                 interval = INTERVAL_UPDATE(deep_idle_res);
411                 } else {
412                         /*
413                          * Restore interval after a refresh.
```



```
414                          */
415                         if (key)
416                                 interval = interval_usr;
417                 }
418         }

420         return (EXIT_SUCCESS);
421 }
```

_____***unchanged_portion_omitted_***

****
5811 Sun Dec 14 23:31:19 2014
new/usr/src/cmd/powertop/common/suggestions.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
****
_____***unchanged_portion_omitted_***

```
 267 void
 268 pt_sugg_as_root(void)
 269 {
 270         pt_sugg_add("Suggestion: run as root to get suggestions"
 271             " for reducing system power consumption",  40, '\0', NULL,
 271             " for reducing system power consumption",  40, NULL, NULL,
 272             NULL);
 273 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   93315 Sun Dec 14 23:31:19 2014
new/usr/src/cmd/svc/svcs/svcs.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 549 static int
 550 get_restarter_time_prop(scf_instance_t *inst, const char *pname,
 551     struct timeval *tvp, int ok_if_empty)
 552 {
 553         int r;

 555         r = inst_get_single_val(inst, SCF_PG_RESTARTER, pname, SCF_TYPE_TIME,
 556             tvp, 0, ok_if_empty ? EMPTY_OK : 0, 0, 1);
 556             tvp, NULL, ok_if_empty ? EMPTY_OK : 0, 0, 1);

 558         return (r == 0 ? 0 : -1);
 559 }
______unchanged_portion_omitted_

1626 /* STIME */
1627 #define STIME_COLUMN_WIDTH              8
1628 #define FORMAT_TIME                     "%k:%M:%S"
1629 #define FORMAT_DATE                     "%b_%d  "
1630 #define FORMAT_YEAR                     "%Y    "

1632 /*
1633  * sprint_stime() will allocate a new buffer and snprintf the services's
1634  * state timestamp.  If the timestamp is unavailable for some reason
1635  * a '-' is given instead.
1636  */
1637 static void
1638 sprint_stime(char **buf, scf_walkinfo_t *wip)
1639 {
1640         int r;
1641         struct timeval tv;
1642         time_t then;
1643         struct tm *tm;
1644         char st_buf[STIME_COLUMN_WIDTH + 1];
1645         size_t newsize = (*buf ? strlen(*buf) : 0) + STIME_COLUMN_WIDTH + 2;
1646         char *newbuf = safe_malloc(newsize);

1648         if (wip->pg == NULL) {
1649                 r = get_restarter_time_prop(wip->inst,
1650                     SCF_PROPERTY_STATE_TIMESTAMP, &tv, 0);
1651         } else {
1652                 r = pg_get_single_val(wip->pg, SCF_PROPERTY_STATE_TIMESTAMP,
1653                     SCF_TYPE_TIME, &tv, 0, 0);
1653                     SCF_TYPE_TIME, &tv, NULL, 0);
1654         }

1656         if (r != 0) {
1657                 /*
1658                  * There's something amiss with our service
1659                  * so we'll print a '-' for STIME.
1660                  */
1661                 (void) snprintf(newbuf, newsize, "%s%-*s", *buf ? *buf : "",
1662                     STIME_COLUMN_WIDTH + 1, "-");
1663         } else {
1664                 /* tv should be valid so we'll format it */
1665                 then = (time_t)tv.tv_sec;

1667                 tm = localtime(&then);
```



```
1668                 /*
1669                  * Print time if started within the past 24 hours, print date
1670                  * if within the past 12 months or, finally, print year if
1671                  * started greater than 12 months ago.
1672                  */
1673                 if (now - then < 24 * 60 * 60) {
1674                         (void) strftime(st_buf, sizeof (st_buf),
1675                             gettext(FORMAT_TIME), tm);
1676                 } else if (now - then < 12 * 30 * 24 * 60 * 60) {
1677                         (void) strftime(st_buf, sizeof (st_buf),
1678                             gettext(FORMAT_DATE), tm);
1679                 } else {
1680                         (void) strftime(st_buf, sizeof (st_buf),
1681                             gettext(FORMAT_YEAR), tm);
1682                 }
1683                 (void) snprintf(newbuf, newsize, "%s%-*s ", *buf ? *buf : "",
1684                     STIME_COLUMN_WIDTH + 1, st_buf);
1685         }
1686         if (*buf)
1687                 free(*buf);
1688         *buf = newbuf;
1689 }

1691 #define STIME_SORTKEY_WIDTH             (sizeof (uint64_t) + sizeof (uint32_t))

1693 /* ARGSUSED */
1694 static void
1695 sortkey_stime(char *buf, int reverse, scf_walkinfo_t *wip)
1696 {
1697         struct timeval tv;
1698         int r;

1700         if (wip->pg == NULL)
1701                 r = get_restarter_time_prop(wip->inst,
1702                     SCF_PROPERTY_STATE_TIMESTAMP, &tv, 0);
1703         else
1704                 r = pg_get_single_val(wip->pg, SCF_PROPERTY_STATE_TIMESTAMP,
1705                     SCF_TYPE_TIME, &tv, 0, 0);
1705                     SCF_TYPE_TIME, &tv, NULL, 0);

1707         if (r == 0) {
1708                 int64_t sec;
1709                 int32_t us;

1711                 /* Stick it straight into the buffer. */
1712                 sec = tv.tv_sec;
1713                 us = tv.tv_usec;

1715                 sec = BE_64(sec);
1716                 us = BE_32(us);
1717                 bcopy(&sec, buf, sizeof (sec));
1718                 bcopy(&us, buf + sizeof (sec), sizeof (us));
1719         } else {
1720                 bzero(buf, STIME_SORTKEY_WIDTH);
1721         }

1723         if (reverse)
1724                 reverse_bytes(buf, STIME_SORTKEY_WIDTH);
1725 }
______unchanged_portion_omitted_

2415 /* ARGSUSED */
2416 static int
2417 print_detailed(void *unused, scf_walkinfo_t *wip)
2418 {
2419         scf_snapshot_t *snap;
```

```
2420         scf_propertygroup_t *rpg;
2421         scf_iter_t *pg_iter;

2423         char *buf;
2424         char *timebuf;
2425         size_t tbsz;
2426         int ret;
2427         uint64_t c;
2428         int temp, perm;
2429         struct timeval tv;
2430         time_t stime;
2431         struct tm *tmp;
2432         int restarter_spec;
2433         int restarter_ret;

2435         const char * const fmt = "%-*s%s\n";

2437         assert(wip->pg == NULL);

2439         rpg = scf_pg_create(h);
2440         if (rpg == NULL)
2441                 scfdie();

2443         if (first_paragraph)
2444                 first_paragraph = 0;
2445         else
2446                 (void) putchar('\n');

2448         buf = safe_malloc(max_scf_fmri_length + 1);

2450         if (scf_instance_to_fmri(wip->inst, buf, max_scf_fmri_length + 1) != -1)
2451                 (void) printf(fmt, DETAILED_WIDTH, "fmri", buf);

2453         if (common_name_buf == NULL)
2454                 common_name_buf = safe_malloc(max_scf_value_length + 1);

2456         if (inst_get_single_val(wip->inst, SCF_PG_TM_COMMON_NAME, locale,
2457             SCF_TYPE_USTRING, common_name_buf, max_scf_value_length, 0, 1, 1)
2458             == 0)
2459                 (void) printf(fmt, DETAILED_WIDTH, gettext("name"),
2460                     common_name_buf);
2461         else if (inst_get_single_val(wip->inst, SCF_PG_TM_COMMON_NAME, "C",
2462             SCF_TYPE_USTRING, common_name_buf, max_scf_value_length, 0, 1, 1)
2463             == 0)
2464                 (void) printf(fmt, DETAILED_WIDTH, gettext("name"),
2465                     common_name_buf);

2467         if (g_zonename != NULL)
2468                 (void) printf(fmt, DETAILED_WIDTH, gettext("zone"), g_zonename);

2470         /*
2471          * Synthesize an 'enabled' property that hides the enabled_ovr
2472          * implementation from the user.  If the service has been temporarily
2473          * set to a state other than its permanent value, alert the user with
2474          * a '(temporary)' message.
2475          */
2476         perm = instance_enabled(wip->inst, B_FALSE);
2477         temp = instance_enabled(wip->inst, B_TRUE);
2478         if (temp != -1) {
2479                 if (temp != perm)
2480                         (void) printf(gettext("%-*s%s (temporary)\n"),
2481                             DETAILED_WIDTH, gettext("enabled"),
2482                             temp ? gettext("true") : gettext("false"));
2483                 else
2484                         (void) printf(fmt, DETAILED_WIDTH,
2485                             gettext("enabled"), temp ? gettext("true") :
```



```
2486                             gettext("false"));
2487         } else if (perm != -1) {
2488                 (void) printf(fmt, DETAILED_WIDTH, gettext("enabled"),
2489                     perm ? gettext("true") : gettext("false"));
2490         }

2492         /*
2493          * Property values may be longer than max_scf_fmri_length, but these
2494          * shouldn't be, so we'll just reuse buf.  The user can use svcprop if
2495          * he suspects something fishy.
2496          */
2497         if (scf_instance_get_pg(wip->inst, SCF_PG_RESTARTER, rpg) != 0) {
2498                 if (scf_error() != SCF_ERROR_NOT_FOUND)
2499                         scfdie();

2501                 scf_pg_destroy(rpg);
2502                 rpg = NULL;
2503         }

2505         if (rpg) {
2506                 if (pg_get_single_val(rpg, scf_property_state, SCF_TYPE_ASTRING,
2507                     buf, max_scf_fmri_length + 1, 0) == 0)
2508                         (void) printf(fmt, DETAILED_WIDTH, gettext("state"),
2509                             buf);

2511                 if (pg_get_single_val(rpg, scf_property_next_state,
2512                     SCF_TYPE_ASTRING, buf, max_scf_fmri_length + 1, 0) == 0)
2513                         (void) printf(fmt, DETAILED_WIDTH,
2514                             gettext("next_state"), buf);

2516                 if (pg_get_single_val(rpg, SCF_PROPERTY_STATE_TIMESTAMP,
2517                     SCF_TYPE_TIME, &tv, 0, 0) == 0) {
2517                     SCF_TYPE_TIME, &tv, NULL, 0) == 0) {
2518                         stime = tv.tv_sec;
2519                         tmp = localtime(&stime);
2520                         for (tbsz = 50; ; tbsz *= 2) {
2521                                 timebuf = safe_malloc(tbsz);
2522                                 if (strftime(timebuf, tbsz, NULL, tmp) != 0)
2523                                         break;
2524                                 free(timebuf);
2525                         }
2526                         (void) printf(fmt, DETAILED_WIDTH,
2527                             gettext("state_time"),
2528                             timebuf);
2529                         free(timebuf);
2530                 }

2532                 if (pg_get_single_val(rpg, SCF_PROPERTY_ALT_LOGFILE,
2533                     SCF_TYPE_ASTRING, buf, max_scf_fmri_length + 1, 0) == 0)
2534                         (void) printf(fmt, DETAILED_WIDTH,
2535                             gettext("alt_logfile"), buf);

2537                 if (pg_get_single_val(rpg, SCF_PROPERTY_LOGFILE,
2538                     SCF_TYPE_ASTRING, buf, max_scf_fmri_length + 1, 0) == 0)
2539                         (void) printf(fmt, DETAILED_WIDTH, gettext("logfile"),
2540                             buf);
2541         }

2543         if (inst_get_single_val(wip->inst, SCF_PG_GENERAL,
2544             SCF_PROPERTY_RESTARTER, SCF_TYPE_ASTRING, buf,
2545             max_scf_fmri_length + 1, 0, 0, 1) == 0)
2546                 (void) printf(fmt, DETAILED_WIDTH, gettext("restarter"), buf);
2547         else
2548                 (void) printf(fmt, DETAILED_WIDTH, gettext("restarter"),
2549                     SCF_SERVICE_STARTD);
```

```
2551         free(buf);

2553         /*
2554          * Use the restarter specific routine to print the ctids, if available.
2555          * If restarter specific action is available and it fails, then die.
2556          */
2557         restarter_ret = ctids_by_restarter(wip, &c, 1, 0,
2558             &restarter_spec, print_ctid_header, print_ctid_detailed);
2559         if (restarter_spec == 1) {
2560                 if (restarter_ret != 0)
2561                         uu_die(gettext("Unable to get restarter for %s"),
2562                             wip->fmri);
2563                 goto restarter_common;
2564         }

2566         if (rpg) {
2567                 scf_iter_t *iter;

2569                 if ((iter = scf_iter_create(h)) == NULL)
2570                         scfdie();

2572                 if (scf_pg_get_property(rpg, scf_property_contract, g_prop) ==
2573                     0) {
2574                         if (scf_property_is_type(g_prop, SCF_TYPE_COUNT) == 0) {

2576                                 /* Callback to print ctid header */
2577                                 print_ctid_header();

2579                                 if (scf_iter_property_values(iter, g_prop) != 0)
2580                                         scfdie();

2582                                 for (;;) {
2583                                         ret = scf_iter_next_value(iter, g_val);
2584                                         if (ret == -1)
2585                                                 scfdie();
2586                                         if (ret == 0)
2587                                                 break;

2589                                         if (scf_value_get_count(g_val, &c) != 0)
2590                                                 scfdie();

2592                                         /* Callback to print contract id. */
2593                                         print_ctid_detailed(c);
2594                                 }

2596                                 (void) putchar('\n');
2597                         } else {
2598                                 if (scf_error() != SCF_ERROR_TYPE_MISMATCH)
2599                                         scfdie();
2600                         }
2601                 } else {
2602                         if (scf_error() != SCF_ERROR_NOT_FOUND)
2603                                 scfdie();
2604                 }

2606                 scf_iter_destroy(iter);
2607         } else {
2608                 if (scf_error() != SCF_ERROR_NOT_FOUND)
2609                         scfdie();
2610         }

2612 restarter_common:
2613         scf_pg_destroy(rpg);

2615         /* Dependencies. */
2616         if ((pg_iter = scf_iter_create(h)) == NULL)
```



```
2617                 scfdie();

2619         snap = get_running_snapshot(wip->inst);

2621         if (scf_iter_instance_pgs_typed_composed(pg_iter, wip->inst, snap,
2622             SCF_GROUP_DEPENDENCY) != SCF_SUCCESS)
2623                 scfdie();

2625         while ((ret = scf_iter_next_pg(pg_iter, g_pg)) == 1)
2626                 print_detailed_dependency(g_pg);
2627         if (ret == -1)
2628                 scfdie();

2630         scf_iter_destroy(pg_iter);

2632         if (opt_processes)
2633                 detailed_list_processes(wip);

2635         /* "application" type property groups */
2636         if (opt_verbose == 1)
2637                 print_application_properties(wip, snap);

2639         scf_snapshot_destroy(snap);

2641         return (0);
2642 }
```

unchanged_portion_omitted_

```
**********************************************************
    8809 Sun Dec 14 23:31:20 2014
new/usr/src/cmd/syseventd/modules/devfsadmd_mod/devfsadmd_mod.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 294 struct slm_mod_ops *
 295 slm_init()
 296 {
 297         char alt_door[MAXPATHLEN];

 299         if (strcmp(root_dir, "") == 0) {
 300                 /* Initialize the private sysevent handle */
 301                 sysevent_hp = sysevent_open_channel_alt(DEVFSADM_SERVICE_DOOR);
 302         } else {

 304                 /* Try alternate door during install time */
 305                 if (snprintf(alt_door, MAXPATHLEN, "%s%s", "/tmp",
 306                     DEVFSADM_SERVICE_DOOR) >= MAXPATHLEN)
 307                         return (NULL);

 309                 sysevent_hp = sysevent_open_channel_alt(alt_door);
 310                 use_alt_root = 1;
 311         }
 312         if (sysevent_hp == NULL) {
 313                 syseventd_print(0, "Unable to allocate sysevent handle"
 314                     " for devfsadm module\n");
 315                 return (NULL);
 316         }

 318         if (sysevent_bind_publisher(sysevent_hp) != 0) {
 319                 if (errno == EBUSY) {
 320                         sysevent_cleanup_publishers(sysevent_hp);
 321                         if (sysevent_bind_publisher(sysevent_hp) != 0) {
 322                                 (void) sysevent_close_channel(sysevent_hp);
 323                                 return (NULL);
 324                         }
 325                 }
 326         }

 328         sysevent_cleanup_subscribers(sysevent_hp);
 329         cleanup = 0;
 330         eventq_head = NULL;
 331         eventq_tail = NULL;

 333         (void) mutex_init(&evq_lock, USYNC_THREAD, NULL);
 334         (void) cond_init(&evq_cv, USYNC_THREAD, NULL);

 336         if (thr_create(NULL, 0, (void *(*)(void *))devfsadmd_deliver_thr,
 336         if (thr_create(NULL, NULL, (void *(*)(void *))devfsadmd_deliver_thr,
 337             NULL, THR_BOUND, &deliver_thr_id) != 0) {
 338                 (void) mutex_destroy(&evq_lock);
 339                 (void) cond_destroy(&evq_cv);
 340                 sysevent_close_channel(sysevent_hp);
 341                 return (NULL);
 342         }

 344         return (&devfsadm_mod_ops);
 345 }
______unchanged_portion_omitted_
```

```
**********************************************************
    6197 Sun Dec 14 23:31:20 2014
new/usr/src/cmd/syseventd/modules/sysevent_reg_mod/sysevent_reg_mod.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 221 static struct slm_mod_ops sysevent_reg_mod_ops = {
 222         SE_MAJOR_VERSION, SE_MINOR_VERSION, SE_MAX_RETRY_LIMIT, deliver_event};

 224 struct slm_mod_ops *
 225 slm_init()
 226 {
 227         cleanup = 0;
 228         sysevent_hp = NULL;

 230         (void) init_channel();

 232         (void) mutex_init(&evq_lock, USYNC_THREAD, NULL);
 233         (void) cond_init(&evq_cv, USYNC_THREAD, NULL);

 235         if (thr_create(NULL, 0, (void *(*)(void *))subscriber_deliver_thr,
 235         if (thr_create(NULL, NULL, (void *(*)(void *))subscriber_deliver_thr,
 236             NULL, 0, &deliver_thr_id) != 0) {
 237                 sysevent_err_print(INIT_SUB_THR_CREATE_ERR, strerror(errno));
 238                 return (NULL);
 239         }

 241         return (&sysevent_reg_mod_ops);
 242 }
______unchanged_portion_omitted_
```

```
**********************************************************
  237697 Sun Dec 14 23:31:20 2014
new/usr/src/cmd/tar/tar.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

2638 /*
2639  *      convtoreg - determines whether the file should be converted to a
2640  *                  regular file when extracted
2641  *
2642  *      Returns 1 when file size > 0 and typeflag is not recognized
2643  *      Otherwise returns 0
2644  */
2645 static int
2646 convtoreg(off_t size)
2647 {
2648         if ((size > 0) && (dblock.dbuf.typeflag != '0') &&
2649             (dblock.dbuf.typeflag != '\0') && (dblock.dbuf.typeflag != '1') &&
2649             (dblock.dbuf.typeflag != NULL) && (dblock.dbuf.typeflag != '1') &&
2650             (dblock.dbuf.typeflag != '2') && (dblock.dbuf.typeflag != '3') &&
2651             (dblock.dbuf.typeflag != '4') && (dblock.dbuf.typeflag != '5') &&
2652             (dblock.dbuf.typeflag != '6') && (dblock.dbuf.typeflag != 'A') &&
2653             (dblock.dbuf.typeflag != 'L') &&
2654             (dblock.dbuf.typeflag != _XATTR_HDRTYPE) &&
2655             (dblock.dbuf.typeflag != 'X')) {
2656                 return (1);
2657         }
2658         return (0);
2659 }
______unchanged_portion_omitted_
2872 #endif

2874 static void
2875 doxtract(char *argv[])
2876 {
2877         struct  stat    xtractbuf;      /* stat on file after extracting */
2878         blkcnt_t blocks;
2879         off_t bytes;
2880         int ofile;
2881         int newfile;                    /* Does the file already exist  */
2882         int xcnt = 0;                   /* count # files extracted */
2883         int fcnt = 0;                   /* count # files in argv list */
2884         int dir;
2885         int dirfd = -1;
2886         int cwd = -1;
2887         int rw_sysattr;
2888         int saveerrno;
2889         uid_t Uid;
2890         char *namep, *dirp, *comp, *linkp; /* for removing absolute paths */
2891         char dirname[PATH_MAX+1];
2892         char templink[PATH_MAX+1];      /* temp link with terminating NULL */
2893         int once = 1;
2894         int error;
2895         int symflag;
2896         int want;
2897         attr_data_t *attrinfo = NULL;   /* attribute info */
2898         acl_t   *aclp = NULL;   /* acl info */
2899         char dot[] = ".";               /* dirp for using realpath */
2900         timestruc_t     time_zero;      /* used for call to doDirTimes */
2901         int             dircreate;
2902         int convflag;
2903         time_zero.tv_sec = 0;
2904         time_zero.tv_nsec = 0;
```



```
2906         /* reset Trusted Extensions variables */
2907         rpath_flag = 0;
2908         lk_rpath_flag = 0;
2909         dir_flag = 0;
2910         mld_flag = 0;
2911         bslundef(&bs_label);
2912         bsllow(&admin_low);
2913         bslhigh(&admin_high);
2914         orig_namep = 0;

2916         dumping = 0;    /* for newvol(), et al:  we are not writing */

2918         Uid = getuid();

2920         for (;;) {
2921                 convflag = 0;
2922                 symflag = 0;
2923                 dir = 0;
2924                 Hiddendir = 0;
2925                 rw_sysattr = 0;
2926                 ofile = -1;

2928                 if (dirfd != -1) {
2929                         (void) close(dirfd);
2930                         dirfd = -1;
2931                 }
2932                 if (ofile != -1) {
2933                         if (close(ofile) != 0)
2934                                 vperror(2, gettext("close error"));
2935                 }

2937 #if defined(O_XATTR)
2938                 if (cwd != -1) {
2939                         rest_cwd(&cwd);
2940                 }
2941 #endif

2943                 /* namep is set by wantit to point to the full name */
2944                 if ((want = wantit(argv, &namep, &dirp, &comp,
2945                     &attrinfo)) == 0) {
2946 #if defined(O_XATTR)
2947                         if (xattrp != NULL) {
2948                                 free(xattrhead);
2949                                 xattrp = NULL;
2950                                 xattr_linkp = NULL;
2951                                 xattrhead = NULL;
2952                         }
2953 #endif
2954                         continue;
2955                 }
2956                 if (want == -1)
2957                         break;

2959 /* Trusted Extensions */
2960                 /*
2961                  * During tar extract (x):
2962                  * If the pathname of the restored file has been
2963                  * reconstructed from the ancillary file,
2964                  * use it to process the normal file.
2965                  */
2966                 if (mld_flag) {         /* Skip over .MLD. directory */
2967                         mld_flag = 0;
2968                         passtape();
2969                         continue;
2970                 }
```

```
2971                orig_namep = namep;     /* save original */
2972                if (rpath_flag) {
2973                        namep = real_path;      /* use zone path */
2974                        comp = real_path;       /* use zone path */
2975                        dirp = dot;             /* work from the top */
2976                        rpath_flag = 0;         /* reset */
2977                }

2979                if (dirfd != -1)
2980                        (void) close(dirfd);

2982                (void) strcpy(&dirname[0], namep);
2983                dircreate = checkdir(&dirname[0]);

2985 #if defined(O_XATTR)
2986                if (xattrp != NULL) {
2987                        int     rc;

2989                        if (((cwd = save_cwd()) == -1) ||
2990                            ((rc = open_attr_dir(comp, dirp, cwd,
2991                            attrinfo)) != ATTR_OK)) {
2992                                if (cwd == -1) {
2993                                        vperror(0, gettext(
2994                                            "unable to save current working "
2995                                            "directory while processing "
2996                                            "attribute %s of %s"),
2997                                            dirp, attrinfo->attr_path);
2998                                } else if (rc != ATTR_SKIP) {
2999                                        (void) fprintf(vfile,
3000                                            gettext("tar: cannot open "
3001                                            "%sattribute %s of file %s: %s\n"),
3002                                            attrinfo->attr_rw_sysattr ? gettext(
3003                                            "system ") : "",
3004                                            comp, dirp, strerror(errno));
3005                                }
3006                                free(xattrhead);
3007                                xattrp = NULL;
3008                                xattr_linkp = NULL;
3009                                xattrhead = NULL;

3011                                passtape();
3012                                continue;
3013                        } else {
3014                                dirfd = attrinfo->attr_parentfd;
3015                                rw_sysattr = attrinfo->attr_rw_sysattr;
3016                        }
3017                } else {
3018                        dirfd = open(dirp, O_RDONLY);
3019                }
3020 #else
3021                dirfd = open(dirp, O_RDONLY);
3022 #endif
3023                if (dirfd == -1) {
3024                        (void) fprintf(vfile, gettext(
3025                            "tar: cannot open %s: %s\n"),
3026                            dirp, strerror(errno));
3027                        passtape();
3028                        continue;
3029                }

3031                if (xhdr_flgs & _X_LINKPATH)
3032                        (void) strcpy(templink, Xtarhdr.x_linkpath);
3033                else {
3034 #if defined(O_XATTR)
3035                        if (xattrp && dblock.dbuf.typeflag == '1') {
3036                                (void) sprintf(templink, "%.*s", NAMSIZ,
```



```
3037                                    xattrp->h_names);
3038                        } else {
3039                                (void) sprintf(templink, "%.*s", NAMSIZ,
3040                                    dblock.dbuf.linkname);
3041                        }
3042 #else
3043                        (void) sprintf(templink, "%.*s", NAMSIZ,
3044                            dblock.dbuf.linkname);
3045 #endif
3046                }

3048                if (Fflag) {
3049                        if (checkf(namep, is_directory(namep), Fflag) == 0) {
3050                                passtape();
3051                                continue;
3052                        }
3053                }

3055                if (checkw('x', namep) == 0) {
3056                        passtape();
3057                        continue;
3058                }
3059                if (once) {
3060                        if (strcmp(dblock.dbuf.magic, magic_type) == 0) {
3061                                if (geteuid() == (uid_t)0) {
3062                                        checkflag = 1;
3063                                        pflag = 1;
3064                                } else {
3065                                        /* get file creation mask */
3066                                        Oumask = umask(0);
3067                                        (void) umask(Oumask);
3068                                }
3069                                once = 0;
3070                        } else {
3071                                if (geteuid() == (uid_t)0) {
3072                                        pflag = 1;
3073                                        checkflag = 2;
3074                                }
3075                                if (!pflag) {
3076                                        /* get file creation mask */
3077                                        Oumask = umask(0);
3078                                        (void) umask(Oumask);
3079                                }
3080                                once = 0;
3081                        }
3082                }

3084 #if defined(O_XATTR)
3085                /*
3086                 * Handle extraction of hidden attr dir.
3087                 * Dir is automatically created, we only
3088                 * need to update mode and perm's.
3089                 */
3090                if ((xattrp != NULL) && Hiddendir == 1) {
3091                        bytes = stbuf.st_size;
3092                        blocks = TBLOCKS(bytes);
3093                        if (vflag) {
3094                                (void) fprintf(vfile,
3095                                    "x %s%s%s, %" FMT_off_t " %s, ", namep,
3096                                    gettext(" attribute "),
3097                                    xattrapath, bytes,
3098                                    gettext("bytes"));
3099                                if (NotTape)
3100                                        (void) fprintf(vfile,
3101                                            "%" FMT_blkcnt_t "K\n", K(blocks));
3102                                else
```

```
3103                                    (void) fprintf(vfile, gettext("%"
3104                                        FMT_blkcnt_t " tape blocks\n"),
3105                                        blocks);
3106                            }
3107
3108                            /*
3109                             * Set the permissions and mode of the attribute
3110                             * unless the attribute is a system attribute (can't
3111                             * successfully do this) or the hidden attribute
3112                             * directory (".") of an attribute (when the attribute
3113                             * is restored, the hidden attribute directory of an
3114                             * attribute is transient).  Note:  when the permissions
3115                             * and mode are set for the hidden attribute directory
3116                             * of a file on a system supporting extended system
3117                             * attributes, even though it returns successfully, it
3118                             * will not have any affect since the attribute
3119                             * directory is transient.
3120                             */
3121                            if (attrinfo->attr_parent == NULL) {
3122                                    if (fchownat(dirfd, ".", stbuf.st_uid,
3123                                        stbuf.st_gid, 0) != 0) {
3124                                            vperror(0, gettext(
3125                                                "%s%s%s: failed to set ownership "
3126                                                "of attribute directory"), namep,
3127                                                gettext(" attribute "), xattrapath);
3128                                    }
3129
3130                                    if (fchmod(dirfd, stbuf.st_mode) != 0) {
3131                                            vperror(0, gettext(
3132                                                "%s%s%s: failed to set permissions "
3133                                                "of attribute directory"), namep,
3134                                                gettext(" attribute "), xattrapath);
3135                                    }
3136                            }
3137                            goto filedone;
3138                    }
3139 #endif
3140
3141                    if (dircreate && (!is_posix || dblock.dbuf.typeflag == '5')) {
3142                            dir = 1;
3143                            if (vflag) {
3144                                    (void) fprintf(vfile, "x %s, 0 %s, ",
3145                                        &dirname[0], gettext("bytes"));
3146                                    if (NotTape)
3147                                            (void) fprintf(vfile, "0K\n");
3148                                    else
3149                                            (void) fprintf(vfile, gettext("%"
3150                                                FMT_blkcnt_t " tape blocks\n"),
3151                                                (blkcnt_t)0);
3152                            }
3153                            goto filedone;
3154                    }
3155
3156                    if (dblock.dbuf.typeflag == '6') {      /* FIFO */
3157                            if (rmdir(namep) < 0) {
3158                                    if (errno == ENOTDIR)
3159                                            (void) unlink(namep);
3160                            }
3161                            linkp = templink;
3162                            if (*linkp !=  '\0') {
3162                            if (*linkp !=  NULL) {
3163                                    if (Aflag && *linkp == '/')
3164                                            linkp++;
3165                                    if (link(linkp, namep) < 0) {
3166                                            (void) fprintf(stderr, gettext(
3167                                                "tar: %s: cannot link\n"), namep);
```



```
3168                                            continue;
3169                                    }
3170                                    if (vflag)
3171                                            (void) fprintf(vfile, gettext(
3172                                                "x %s linked to %s\n"), namep,
3173                                                linkp);
3174                                    xcnt++;  /* increment # files extracted */
3175                                    continue;
3176                            }
3177                            if (mknod(namep, (int)(Gen.g_mode|S_IFIFO),
3178                                (int)Gen.g_devmajor) < 0) {
3179                                    vperror(0, gettext("%s: mknod failed"), namep);
3180                                    continue;
3181                            }
3182                            bytes = stbuf.st_size;
3183                            blocks = TBLOCKS(bytes);
3184                            if (vflag) {
3185                                    (void) fprintf(vfile, "x %s, %" FMT_off_t
3186                                        " %s, ", namep, bytes, gettext("bytes"));
3187                                    if (NotTape)
3188                                            (void) fprintf(vfile, "%" FMT_blkcnt_t
3189                                                "K\n", K(blocks));
3190                                    else
3191                                            (void) fprintf(vfile, gettext("%"
3192                                                FMT_blkcnt_t " tape blocks\n"),
3193                                                blocks);
3194                            }
3195                            goto filedone;
3196                    }
3197                    if (dblock.dbuf.typeflag == '3' && !Uid) { /* CHAR SPECIAL */
3198                            if (rmdir(namep) < 0) {
3199                                    if (errno == ENOTDIR)
3200                                            (void) unlink(namep);
3201                            }
3202                            linkp = templink;
3203                            if (*linkp != '\0') {
3203                            if (*linkp != NULL) {
3204                                    if (Aflag && *linkp == '/')
3205                                            linkp++;
3206                                    if (link(linkp, namep) < 0) {
3207                                            (void) fprintf(stderr, gettext(
3208                                                "tar: %s: cannot link\n"), namep);
3209                                            continue;
3210                                    }
3211                                    if (vflag)
3212                                            (void) fprintf(vfile, gettext(
3213                                                "x %s linked to %s\n"), namep,
3214                                                linkp);
3215                                    xcnt++;  /* increment # files extracted */
3216                                    continue;
3217                            }
3218                            if (mknod(namep, (int)(Gen.g_mode|S_IFCHR),
3219                                (int)makedev(Gen.g_devmajor, Gen.g_devminor)) < 0) {
3220                                    vperror(0, gettext(
3221                                        "%s: mknod failed"), namep);
3222                                    continue;
3223                            }
3224                            bytes = stbuf.st_size;
3225                            blocks = TBLOCKS(bytes);
3226                            if (vflag) {
3227                                    (void) fprintf(vfile, "x %s, %" FMT_off_t
3228                                        " %s, ", namep, bytes, gettext("bytes"));
3229                                    if (NotTape)
3230                                            (void) fprintf(vfile, "%" FMT_blkcnt_t
3231                                                "K\n", K(blocks));
3232                                    else
```

```
3233                                    (void) fprintf(vfile, gettext("%"
3234                                        FMT_blkcnt_t " tape blocks\n"),
3235                                        blocks);
3236                            }
3237                            goto filedone;
3238                    } else if (dblock.dbuf.typeflag == '3' && Uid) {
3239                            (void) fprintf(stderr, gettext(
3240                                "Can't create special %s\n"), namep);
3241                            continue;
3242                    }

3244                    /* BLOCK SPECIAL */

3246                    if (dblock.dbuf.typeflag == '4' && !Uid) {
3247                            if (rmdir(namep) < 0) {
3248                                    if (errno == ENOTDIR)
3249                                            (void) unlink(namep);
3250                            }
3251                            linkp = templink;
3252                            if (*linkp != '\0') {
3252                            if (*linkp != NULL) {
3253                                    if (Aflag && *linkp == '/')
3254                                            linkp++;
3255                                    if (link(linkp, namep) < 0) {
3256                                            (void) fprintf(stderr, gettext(
3257                                                "tar: %s: cannot link\n"), namep);
3258                                            continue;
3259                                    }
3260                                    if (vflag)
3261                                            (void) fprintf(vfile, gettext(
3262                                                "x %s linked to %s\n"), namep,
3263                                                linkp);
3264                                    xcnt++;  /* increment # files extracted */
3265                                    continue;
3266                            }
3267                            if (mknod(namep, (int)(Gen.g_mode|S_IFBLK),
3268                                (int)makedev(Gen.g_devmajor, Gen.g_devminor)) < 0) {
3269                                    vperror(0, gettext("%s: mknod failed"), namep);
3270                                    continue;
3271                            }
3272                            bytes = stbuf.st_size;
3273                            blocks = TBLOCKS(bytes);
3274                            if (vflag) {
3275                                    (void) fprintf(vfile, gettext("x %s, %"
3276                                        FMT_off_t " bytes, "), namep, bytes);
3277                                    if (NotTape)
3278                                            (void) fprintf(vfile, "%" FMT_blkcnt_t
3279                                                "K\n", K(blocks));
3280                                    else
3281                                            (void) fprintf(vfile, gettext("%"
3282                                                FMT_blkcnt_t " tape blocks\n"),
3283                                                blocks);
3284                            }
3285                            goto filedone;
3286                    } else if (dblock.dbuf.typeflag == '4' && Uid) {
3287                            (void) fprintf(stderr,
3288                                gettext("Can't create special %s\n"), namep);
3289                            continue;
3290                    }
3291                    if (dblock.dbuf.typeflag == '2') {      /* symlink */
3292                            if ((Tflag) && (lk_rpath_flag == 1))
3293                                    linkp = lk_real_path;
3294                            else
3295                                    linkp = templink;
3296                            if (Aflag && *linkp == '/')
3297                                    linkp++;
```



```
3298                            if (rmdir(namep) < 0) {
3299                                    if (errno == ENOTDIR)
3300                                            (void) unlink(namep);
3301                            }
3302                            if (symlink(linkp, namep) < 0) {
3303                                    vperror(0, gettext("%s: symbolic link failed"),
3304                                        namep);
3305                                    continue;
3306                            }
3307                            if (vflag)
3308                                    (void) fprintf(vfile, gettext(
3309                                        "x %s symbolic link to %s\n"),
3310                                        namep, linkp);

3312                            symflag = AT_SYMLINK_NOFOLLOW;
3313                            goto filedone;
3314                    }
3315                    if (dblock.dbuf.typeflag == '1') {
3316                            linkp = templink;
3317                            if (Aflag && *linkp == '/')
3318                                    linkp++;
3319                            if (unlinkat(dirfd, comp, AT_REMOVEDIR) < 0) {
3320                                    if (errno == ENOTDIR)
3321                                            (void) unlinkat(dirfd, comp, 0);
3322                            }
3323 #if defined(O_XATTR)
3324                            if (xattrp && xattr_linkp) {
3325                                    if (fchdir(dirfd) < 0) {
3326                                            vperror(0, gettext(
3327                                                "Cannot fchdir to attribute "
3328                                                "directory %s"),
3329                                                (attrinfo->attr_parent == NULL) ?
3330                                                dirp : attrinfo->attr_parent);
3331                                            exit(1);
3332                                    }

3334                                    error = link(xattr_linkaname, xattrapath);
3335                            } else {
3336                                    error = link(linkp, namep);
3337                            }
3338 #else
3339                            error = link(linkp, namep);
3340 #endif

3342                            if (error < 0) {
3343                                    (void) fprintf(stderr, gettext(
3344                                        "tar: %s%s%s: cannot link\n"),
3345                                        namep, (xattr_linkp != NULL) ?
3346                                        gettext(" attribute ") : "",
3347                                        (xattr_linkp != NULL) ?
3348                                        xattrapath : "");
3349                                    continue;
3350                            }
3351                            if (vflag)
3352                                    (void) fprintf(vfile, gettext(
3353                                        "x %s%s%s linked to %s%s%s\n"), namep,
3354                                        (xattr_linkp != NULL) ?
3355                                        gettext(" attribute ") : "",
3356                                        (xattr_linkp != NULL) ?
3357                                        xattr_linkaname : "",
3358                                        linkp,
3359                                        (xattr_linkp != NULL) ?
3360                                        gettext(" attribute ") : "",
3361                                        (xattr_linkp != NULL) ? xattrapath : "");
3362                            xcnt++;         /* increment # files extracted */
3363 #if defined(O_XATTR)
```

```
3364                        if (xattrp != NULL) {
3365                                free(xattrhead);
3366                                xattrp = NULL;
3367                                xattr_linkp = NULL;
3368                                xattrhead = NULL;
3369                        }
3370 #endif
3371                        continue;
3372                }

3374                /* REGULAR FILES */

3376                if (convtoreg(stbuf.st_size)) {
3377                        convflag = 1;
3378                        if (errflag) {
3379                                (void) fprintf(stderr, gettext(
3380                                    "tar: %s: typeflag '%c' not recognized\n"),
3381                                    namep, dblock.dbuf.typeflag);
3382                                done(1);
3383                        } else {
3384                                (void) fprintf(stderr, gettext(
3385                                    "tar: %s: typeflag '%c' not recognized, "
3386                                    "converting to regular file\n"), namep,
3387                                    dblock.dbuf.typeflag);
3388                                Errflg = 1;
3389                        }
3390                }
3391                if (dblock.dbuf.typeflag == '0' ||
3392                    dblock.dbuf.typeflag == '\0' || convflag) {
3392                    dblock.dbuf.typeflag == NULL || convflag) {
3393                        delete_target(dirfd, comp, namep);
3394                        linkp = templink;
3395                        if (*linkp != '\0') {
3395                        if (*linkp != NULL) {
3396                                if (Aflag && *linkp == '/')
3397                                        linkp++;
3398                                if (link(linkp, comp) < 0) {
3399                                        (void) fprintf(stderr, gettext(
3400                                            "tar: %s: cannot link\n"), namep);
3401                                        continue;
3402                                }
3403                                if (vflag)
3404                                        (void) fprintf(vfile, gettext(
3405                                            "x %s linked to %s\n"), comp,
3406                                            linkp);
3407                                xcnt++;  /* increment # files extracted */
3408 #if defined(O_XATTR)
3409                                if (xattrp != NULL) {
3410                                        free(xattrhead);
3411                                        xattrp = NULL;
3412                                        xattr_linkp = NULL;
3413                                        xattrhead = NULL;
3414                                }
3415 #endif
3416                                continue;
3417                        }
3418                newfile = ((fstatat(dirfd, comp,
3419                    &xtractbuf, 0) == -1) ? TRUE : FALSE);
3420                ofile = openat(dirfd, comp, O_RDWR|O_CREAT|O_TRUNC,
3421                    stbuf.st_mode & MODEMASK);
3422                saveerrno = errno;

3424 #if defined(O_XATTR)
3425                if (xattrp != NULL) {
3426                        if (ofile < 0) {
3427                                ofile = retry_open_attr(dirfd, cwd,
```



```
3428                                    dirp, attrinfo->attr_parent, comp,
3429                                    O_RDWR|O_CREAT|O_TRUNC,
3430                                    stbuf.st_mode & MODEMASK);
3431                        }
3432                }
3433 #endif
3434                if (ofile < 0) {
3435                        errno = saveerrno;
3436                        (void) fprintf(stderr, gettext(
3437                            "tar: %s%s%s%s - cannot create\n"),
3438                            (xattrp == NULL) ? "" : (rw_sysattr ?
3439                            gettext("system attribute ") :
3440                            gettext("attribute ")),
3441                            (xattrp == NULL) ? "" : xattrapath,
3442                            (xattrp == NULL) ? "" : gettext(" of "),
3443                            (xattrp == NULL) ? comp : namep);
3444                        if (errflag)
3445                                done(1);
3446                        else
3447                                Errflg = 1;
3448 #if defined(O_XATTR)
3449                        if (xattrp != NULL) {
3450                                dblock.dbuf.typeflag = _XATTR_HDRTYPE;
3451                                free(xattrhead);
3452                                xattrp = NULL;
3453                                xattr_linkp = NULL;
3454                                xattrhead = NULL;
3455                        }
3456 #endif
3457                        passtape();
3458                        continue;
3459                }

3461                if (Tflag && (check_ext_attr(namep) == 0)) {
3462                        if (errflag)
3463                                done(1);
3464                        else
3465                                Errflg = 1;
3466                        passtape();
3467                        continue;
3468                }

3470                if (extno != 0) {       /* file is in pieces */
3471                        if (extotal < 1 || extotal > MAXEXT)
3472                                (void) fprintf(stderr, gettext(
3473                                    "tar: ignoring bad extent info for "
3474                                    "%s%s%s%s\n"),
3475                                    (xattrp == NULL) ? "" : (rw_sysattr ?
3476                                    gettext("system attribute ") :
3477                                    gettext("attribute ")),
3478                                    (xattrp == NULL) ? "" : xattrapath,
3479                                    (xattrp == NULL) ? "" : gettext(" of "),
3480                                    (xattrp == NULL) ? comp : namep);
3481                        else {
3482                                /* extract it */
3483                                (void) xsfile(rw_sysattr, ofile);
3484                        }
3485                }
3486                extno = 0;      /* let everyone know file is not split */
3487                bytes = stbuf.st_size;
3488                blocks = TBLOCKS(bytes);
3489                if (vflag) {
3490                        (void) fprintf(vfile,
3491                            "x %s%s%s, %" FMT_off_t " %s, ",
3492                            (xattrp == NULL) ? "" : dirp,
3493                            (xattrp == NULL) ? "" : (rw_sysattr ?
```

```
3494                            gettext(" system attribute ") :
3495                            gettext(" attribute ")),
3496                            (xattrp == NULL) ? namep : xattrapath, bytes,
3497                            gettext("bytes"));
3498                        if (NotTape)
3499                                (void) fprintf(vfile, "%" FMT_blkcnt_t "K\n",
3500                                    K(blocks));
3501                        else
3502                                (void) fprintf(vfile, gettext("%"
3503                                    FMT_blkcnt_t " tape blocks\n"), blocks);
3504                }

3506                if (xblocks(rw_sysattr, bytes, ofile) != 0) {
3507 #if defined(O_XATTR)
3508                        if (xattrp != NULL) {
3509                                free(xattrhead);
3510                                xattrp = NULL;
3511                                xattr_linkp = NULL;
3512                                xattrhead = NULL;
3513                        }
3514 #endif
3515                        continue;
3516                }
3517 filedone:
3518                if (mflag == 0 && !symflag) {
3519                        if (dir)
3520                                doDirTimes(namep, stbuf.st_mtim);

3522                        else
3523 #if defined(O_XATTR)
3524                                if (xattrp != NULL) {
3525                                        /*
3526                                         * Set the time on the attribute unless
3527                                         * the attribute is a system attribute
3528                                         * (can't successfully do this) or the
3529                                         * hidden attribute directory, "." (the
3530                                         * time on the hidden attribute
3531                                         * directory will be updated when
3532                                         * attributes are restored, otherwise
3533                                         * it's transient).
3534                                         */
3535                                        if (!rw_sysattr && (Hiddendir == 0)) {
3536                                                setPathTimes(dirfd, comp,
3537                                                    stbuf.st_mtim);
3538                                        }
3539                                } else
3540                                        setPathTimes(dirfd, comp,
3541                                            stbuf.st_mtim);
3542 #else
3543                                setPathTimes(dirfd, comp, stbuf.st_mtim);
3544 #endif
3545                }

3547                /* moved this code from above */
3548                if (pflag && !symflag && Hiddendir == 0) {
3549                        if (xattrp != NULL)
3550                                (void) fchmod(ofile, stbuf.st_mode & MODEMASK);
3551                        else
3552                                (void) chmod(namep, stbuf.st_mode & MODEMASK);
3553                }


3556                /*
3557                 * Because ancillary file preceeds the normal file,
3558                 * acl info may have been retrieved (in aclp).
3559                 * All file types are directed here (go filedone).
```



```
3560                 * Always restore ACLs if there are ACLs.
3561                 */
3562                if (aclp != NULL) {
3563                        int ret;

3565 #if defined(O_XATTR)
3566                        if (xattrp != NULL) {
3567                                if (Hiddendir)
3568                                        ret = facl_set(dirfd, aclp);
3569                                else
3570                                        ret = facl_set(ofile, aclp);
3571                        } else {
3572                                ret = acl_set(namep, aclp);
3573                        }
3574 #else
3575                        ret = acl_set(namep, aclp);
3576 #endif
3577                        if (ret < 0) {
3578                                if (pflag) {
3579                                        (void) fprintf(stderr, gettext(
3580                                            "%s%s%s%s: failed to set acl "
3581                                            "entries\n"), namep,
3582                                            (xattrp == NULL) ? "" :
3583                                            (rw_sysattr ? gettext(
3584                                            " system attribute ") :
3585                                            gettext(" attribute ")),
3586                                            (xattrp == NULL) ? "" :
3587                                            xattrapath);
3588                                }
3589                                /* else: silent and continue */
3590                        }
3591                        acl_free(aclp);
3592                        aclp = NULL;
3593                }

3595                if (!oflag)
3596                        /* set file ownership */
3597                        resugname(dirfd, comp, symflag);

3599                if (pflag && newfile == TRUE && !dir &&
3600                    (dblock.dbuf.typeflag == '0' ||
3601                    dblock.dbuf.typeflag == '\0' ||
3601                    dblock.dbuf.typeflag == NULL ||
3602                    convflag || dblock.dbuf.typeflag == '1')) {
3603                        if (fstat(ofile, &xtractbuf) == -1)
3604                                (void) fprintf(stderr, gettext(
3605                                    "tar: cannot stat extracted file "
3606                                    "%s%s%s%s\n"),
3607                                    (xattrp == NULL) ? "" : (rw_sysattr ?
3608                                    gettext("system attribute ") :
3609                                    gettext("attribute ")),
3610                                    (xattrp == NULL) ? "" : xattrapath,
3611                                    (xattrp == NULL) ? "" :
3612                                    gettext(" of "), namep);

3614                        else if ((xtractbuf.st_mode & (MODEMASK & ~S_IFMT))
3615                            != (stbuf.st_mode & (MODEMASK & ~S_IFMT))) {
3616                                (void) fprintf(stderr, gettext(
3617                                    "tar: warning - file permissions have "
3618                                    "changed for %s%s%s%s (are 0%o, should be "
3619                                    "0%o)\n"),
3620                                    (xattrp == NULL) ? "" : (rw_sysattr ?
3621                                    gettext("system attribute ") :
3622                                    gettext("attribute ")),
3623                                    (xattrp == NULL) ? "" : xattrapath,
3624                                    (xattrp == NULL) ? "" :
```

```
3625                                    gettext(" of "), namep,
3626                                    xtractbuf.st_mode, stbuf.st_mode);

3628                            }
3629                    }
3630 #if defined(O_XATTR)
3631                    if (xattrp != NULL) {
3632                            free(xattrhead);
3633                            xattrp = NULL;
3634                            xattr_linkp = NULL;
3635                            xattrhead = NULL;
3636                    }
3637 #endif

3639                    if (ofile != -1) {
3640                            (void) close(dirfd);
3641                            dirfd = -1;
3642                            if (close(ofile) != 0)
3643                                    vperror(2, gettext("close error"));
3644                            ofile = -1;
3645                    }
3646                    xcnt++;                 /* increment # files extracted */
3647                    }

3649                    /*
3650                     * Process ancillary file.
3651                     *
3652                     */

3654                    if (dblock.dbuf.typeflag == 'A') {      /* acl info */
3655                            char    buf[TBLOCK];
3656                            char    *secp;
3657                            char    *tp;
3658                            int     attrsize;
3659                            int     cnt;

3661                            /* reset Trusted Extensions flags */
3662                            dir_flag = 0;
3663                            mld_flag = 0;
3664                            lk_rpath_flag = 0;
3665                            rpath_flag = 0;

3667                            if (pflag) {
3668                                    bytes = stbuf.st_size;
3669                                    if ((secp = malloc((int)bytes)) == NULL) {
3670                                            (void) fprintf(stderr, gettext(
3671                                                "Insufficient memory for acl\n"));
3672                                            passtape();
3673                                            continue;
3674                                    }
3675                                    tp = secp;
3676                                    blocks = TBLOCKS(bytes);

3678                                    /*
3679                                     * Display a line for each ancillary file.
3680                                     */
3681                                    if (vflag && Tflag)
3682                                            (void) fprintf(vfile, "x %s(A), %"
3683                                                FMT_blkcnt_t " %s, %"
3684                                                FMT_blkcnt_t " %s\n",
3685                                                namep, bytes, gettext("bytes"),
3686                                                blocks, gettext("tape blocks"));

3688                                    while (blocks-- > 0) {
3689                                            readtape(buf);
3690                                            if (bytes <= TBLOCK) {
```



```
3691                                                    (void) memcpy(tp, buf,
3692                                                        (size_t)bytes);
3693                                                    break;
3694                                            } else {
3695                                                    (void) memcpy(tp, buf,
3696                                                        TBLOCK);
3697                                                    tp += TBLOCK;
3698                                            }
3699                                            bytes -= TBLOCK;
3700                                    }
3701                                    bytes = stbuf.st_size;
3702                                    /* got all attributes in secp */
3703                                    tp = secp;
3704                                    do {
3705                                            attr = (struct sec_attr *)tp;
3706                                            switch (attr->attr_type) {
3707                                            case UFSD_ACL:
3708                                            case ACE_ACL:
3709                                                    (void) sscanf(attr->attr_len,
3710                                                        "%7o",
3711                                                        (uint_t *)
3712                                                        &cnt);
3713                                                    /* header is 8 */
3714                                                    attrsize = 8 + (int)strlen(
3715                                                        &attr->attr_info[0]) + 1;
3716                                                    error =
3717                                                        acl_fromtext(
3718                                                        &attr->attr_info[0], &aclp);

3720                                                    if (error != 0) {
3721                                                            (void) fprintf(stderr,
3722                                                                gettext(
3723                                                                "aclfromtext "
3724                                                                "failed: %s\n"),
3725                                                                acl_strerror(
3726                                                                error));
3727                                                            bytes -= attrsize;
3728                                                            break;
3729                                                    }
3730                                                    if (acl_cnt(aclp) != cnt) {
3731                                                            (void) fprintf(stderr,
3732                                                                gettext(
3733                                                                "aclcnt error\n"));
3734                                                            bytes -= attrsize;
3735                                                            break;
3736                                                    }
3737                                                    bytes -= attrsize;
3738                                                    break;

3740                                            /* Trusted Extensions */

3742                                            case DIR_TYPE:
3743                                            case LBL_TYPE:
3744                                            case APRIV_TYPE:
3745                                            case FPRIV_TYPE:
3746                                            case COMP_TYPE:
3747                                            case LK_COMP_TYPE:
3748                                            case ATTR_FLAG_TYPE:
3749                                                    attrsize =
3750                                                        sizeof (struct sec_attr) +
3751                                                        strlen(&attr->attr_info[0]);
3752                                                    bytes -= attrsize;
3753                                                    if (Tflag)
3754                                                            extract_attr(&namep,
3755                                                                attr);
3756                                                    break;
```

```
3758                                        default:
3759                                                (void) fprintf(stderr, gettext(
3760                                                    "unrecognized attr"
3761                                                    " type\n"));
3762                                                bytes = (off_t)0;
3763                                                break;
3764                                        }

3766                                        /* next attributes */
3767                                        tp += attrsize;
3768                                } while (bytes != 0);
3769                                free(secp);
3770                        } else {
3771                                passtape();
3772                        }
3773                } /* acl */

3775        } /* for */

3777        /*
3778         *  Ensure that all the directories still on the directory stack
3779         *  get their modification times set correctly by flushing the
3780         *  stack.
3781         */

3783        doDirTimes(NULL, time_zero);

3785 #if defined(O_XATTR)
3786                if (xattrp != NULL) {
3787                        free(xattrhead);
3788                        xattrp = NULL;
3789                        xattr_linkp = NULL;
3790                        xattrhead = NULL;
3791                }
3792 #endif

3794        /*
3795         * Check if the number of files extracted is different from the
3796         * number of files listed on the command line
3797         */
3798        if (fcnt > xcnt) {
3799                (void) fprintf(stderr,
3800                    gettext("tar: %d file(s) not extracted\n"),
3801                    fcnt-xcnt);
3802                Errflg = 1;
3803        }
3804 }
```

unchanged_portion_omitted_

```
5474 /*
5475  * Add a file name to the the specified table, if the file name has any
5476  * trailing '/'s then delete them before inserting into the table
5477  */

5479 static void
5480 add_file_to_table(file_list_t *table[], char *str)
5481 {
5482        char    name[PATH_MAX + 1];
5483        unsigned int h;
5484        file_list_t     *exp;

5486        (void) strcpy(name, str);
5487        while (name[strlen(name) - 1] == '/') {
5488                name[strlen(name) - 1] = '\0';
```



```
5488                name[strlen(name) - 1] = NULL;
5489        }

5491        h = hash(name);
5492        if ((exp = (file_list_t *)calloc(sizeof (file_list_t),
5493            sizeof (char))) == NULL) {
5494                (void) fprintf(stderr, gettext(
5495                    "tar: out of memory, exclude/include table(entry)\n"));
5496                exit(1);
5497        }

5499        if ((exp->name = strdup(name)) == NULL) {
5500                (void) fprintf(stderr, gettext(
5501                    "tar: out of memory, exclude/include table(file name)\n"));
5502                exit(1);
5503        }

5505        exp->next = table[h];
5506        table[h] = exp;
5507 }


5510 /*
5511  * See if a file name or any of the file's parent directories is in the
5512  * specified table, if the file name has any trailing '/'s then delete
5513  * them before searching the table
5514  */

5516 static int
5517 is_in_table(file_list_t *table[], char *str)
5518 {
5519        char    name[PATH_MAX + 1];
5520        unsigned int    h;
5521        file_list_t     *exp;
5522        char    *ptr;

5524        (void) strcpy(name, str);
5525        while (name[strlen(name) - 1] == '/') {
5526                name[strlen(name) - 1] = '\0';
5526                name[strlen(name) - 1] = NULL;
5527        }

5529        /*
5530         * check for the file name in the passed list
5531         */
5532        h = hash(name);
5533        exp = table[h];
5534        while (exp != NULL) {
5535                if (strcmp(name, exp->name) == 0) {
5536                        return (1);
5537                }
5538                exp = exp->next;
5539        }

5541        /*
5542         * check for any parent directories in the file list
5543         */
5544        while ((ptr = strrchr(name, '/'))) {
5545                *ptr = '\0';
5545                *ptr = NULL;
5546                h = hash(name);
5547                exp = table[h];
5548                while (exp != NULL) {
5549                        if (strcmp(name, exp->name) == 0) {
5550                                return (1);
5551                        }
```

```
5552                        exp = exp->next;
5553                }
5554        }

5556        return (0);
5557 }
```

*_____unchanged_portion_omitted_*

```
6596 /*
6597  * If hflag is set then delete the symbolic link's target.
6598  * If !hflag then delete the target.
6599  */

6601 static void
6602 delete_target(int fd, char *comp, char *namep)
6603 {
6604        struct  stat    xtractbuf;
6605        char buf[PATH_MAX + 1];
6606        int n;


6609        if (unlinkat(fd, comp, AT_REMOVEDIR) < 0) {
6610                if (errno == ENOTDIR && !hflag) {
6611                        (void) unlinkat(fd, comp, 0);
6612                } else if (errno == ENOTDIR && hflag) {
6613                        if (!lstat(namep, &xtractbuf)) {
6614                                if ((xtractbuf.st_mode & S_IFMT) != S_IFLNK) {
6615                                        (void) unlinkat(fd, comp, 0);
6616                                } else if ((n = readlink(namep, buf,
6617                                    PATH_MAX)) != -1) {
6618                                        buf[n] = '\0';
6618                                        buf[n] = (char)NULL;
6619                                        (void) unlinkat(fd, buf,
6620                                            AT_REMOVEDIR);
6621                                        if (errno == ENOTDIR)
6622                                                (void) unlinkat(fd, buf, 0);
6623                                } else {
6624                                        (void) unlinkat(fd, comp, 0);
6625                                }
6626                        } else {
6627                                (void) unlinkat(fd, comp, 0);
6628                        }
6629                }
6630        }
6631 }
```

*_____unchanged_portion_omitted_*

****
10828 Sun Dec 14 23:31:22 2014
new/usr/src/cmd/tnf/tnfxtract/tnfxtract.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
****
_____***unchanged_portion_omitted_***

```
 109 static void
 110 dumpfile_init()

 112 {
 113         kvm_p = kvm_open(namelist, dumpfile, NULL, O_RDONLY, program_name);
 114         if (kvm_p == NULL) {
 115                 /* kvm_open prints an error message */
 116                 exit(1);
 117         }
 118         if (kvm_nlist(kvm_p, kvm_syms) != 0) {
 119                 (void) fprintf(stderr, gettext(
 120                         "Symbol lookup error in %s\n"), namelist);
 121                 exit(1);
 122         }
 123         if (kvm_read(kvm_p, kvm_syms[0].n_value, (char *) &dump_bufaddr,
 124             sizeof (dump_bufaddr)) != sizeof (dump_bufaddr) ||
 125             kvm_read(kvm_p, kvm_syms[1].n_value, (char *) &tnf_bufsize,
 126             sizeof (tnf_bufsize)) != sizeof (tnf_bufsize)) {
 127                 (void) fprintf(stderr, gettext(
 128                         "kvm_read error in %s\n"), dumpfile);
 129                 exit(1);
 130         }
 131         if (dump_bufaddr == (uintptr_t)NULL || tnf_bufsize == 0) {
 131         if (dump_bufaddr == NULL || tnf_bufsize == 0) {
 132                 (void) fprintf(stderr, gettext(
 133                         "No trace data available in the kernel.\n"));
 134                 exit(1);
 135         }
 136 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2586 Sun Dec 14 23:31:22 2014
new/usr/src/head/iso/stddef_iso.h
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License, Version 1.0 only
 * (the "License").  You may not use this file except in compliance
 * with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*      Copyright (c) 1988 AT&T */
/*        All Rights Reserved   */


/*
 * Copyright 1999-2003 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2014 PALO, Richard.
 */

/*
 * An application should not include this header directly.  Instead it
 * should be included only through the inclusion of other Sun headers.
 *
 * The contents of this header is limited to identifiers specified in the
 * C Standard.  Any new identifiers specified in future amendments to the
 * C Standard must be placed in this header.  If these new identifiers
 * are required to also be in the C++ Standard "std" namespace, then for
 * anything other than macro definitions, corresponding "using" directives
 * must also be added to <stddef.h.h>.
 */

#ifndef _ISO_STDDEF_ISO_H
#define _ISO_STDDEF_ISO_H

#pragma ident   "%Z%%M% %I%     %E% SMI" /* SVr4.0 1.5 */

#include <sys/isa_defs.h>
#include <sys/null.h>

#ifdef  __cplusplus
extern "C" {
#endif

#if __cplusplus >= 199711L
```



```
namespace std {
#endif

#ifndef NULL
#if defined(_LP64)
#define NULL    0L
#else
#define NULL    0
#endif
#endif

#if !defined(_PTRDIFF_T) || __cplusplus >= 199711L
#define _PTRDIFF_T
#if defined(_LP64) || defined(_I32LPx)
typedef long    ptrdiff_t;              /* pointer difference */
#else
typedef int     ptrdiff_t;              /* (historical version) */
#endif
#endif  /* !_PTRDIFF_T */

#if !defined(_SIZE_T) || __cplusplus >= 199711L
#define _SIZE_T
#if defined(_LP64) || defined(_I32LPx)
typedef unsigned long   size_t;         /* size of something in bytes */
#else
typedef unsigned int    size_t;         /* (historical version) */
#endif
#endif  /* !_SIZE_T */

#if __cplusplus >= 199711L
}
_____unchanged_portion_omitted_
```

```
**********************************************************
   26681 Sun Dec 14 23:31:22 2014
new/usr/src/head/unistd.h
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
```

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

/*
 * Copyright 2014 Garrett D'Amore <garrett@damore.org>
 * Copyright (c) 2013 Gary Mills
 *
 * Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
 */

/*      Copyright (c) 1988 AT&T */
/*        All Rights Reserved   */

/* Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved. */

/*
 * Copyright 2014 PALO, Richard.
 */

#ifndef _UNISTD_H
#define _UNISTD_H

#include <sys/feature_tests.h>

#include <sys/null.h>
#include <sys/types.h>
#include <sys/unistd.h>

#ifdef  __cplusplus
extern "C" {
#endif

/* Symbolic constants for the "access" routine: */
#define R_OK    4       /* Test for Read permission */
#define W_OK    2       /* Test for Write permission */
#define X_OK    1       /* Test for eXecute permission */
#define F_OK    0       /* Test for existence of File */

#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
#define F_ULOCK 0       /* Unlock a previously locked region */
#define F_LOCK  1       /* Lock a region for exclusive use */
```



```
#define F_TLOCK 2       /* Test and lock a region for exclusive use */
#define F_TEST  3       /* Test a region for other processes locks */
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */

/* Symbolic constants for the "lseek" routine: */

#ifndef SEEK_SET
#define SEEK_SET        0       /* Set file pointer to "offset" */
#endif

#ifndef SEEK_CUR
#define SEEK_CUR        1       /* Set file pointer to current plus "offset" */
#endif

#ifndef SEEK_END
#define SEEK_END        2       /* Set file pointer to EOF plus "offset" */
#endif

#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
#ifndef SEEK_DATA
#define SEEK_DATA       3       /* Set file pointer to next data past offset */
#endif

#ifndef SEEK_HOLE
#define SEEK_HOLE       4       /* Set file pointer to next hole past offset */
#endif
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
/* Path names: */
#define GF_PATH "/etc/group"    /* Path name of the "group" file */
#define PF_PATH "/etc/passwd"   /* Path name of the "passwd" file */
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

/*
 * compile-time symbolic constants,
 * Support does not mean the feature is enabled.
 * Use pathconf/sysconf to obtain actual configuration value.
 */

/* Values unchanged in UNIX 03 */
#define _POSIX_ASYNC_IO                 1
#define _POSIX_JOB_CONTROL              1
#define _POSIX_SAVED_IDS                1
#define _POSIX_SYNC_IO                  1

/*
 * POSIX.1b compile-time symbolic constants.
 */
#if defined(_XPG6)
#define _POSIX_ASYNCHRONOUS_IO          200112L
#define _POSIX_FSYNC                    200112L
#define _POSIX_MAPPED_FILES             200112L
#define _POSIX_MEMLOCK                  200112L
#define _POSIX_MEMLOCK_RANGE            200112L
#define _POSIX_MEMORY_PROTECTION        200112L
#define _POSIX_MESSAGE_PASSING          200112L
#define _POSIX_PRIORITY_SCHEDULING      200112L
#define _POSIX_REALTIME_SIGNALS         200112L
#define _POSIX_SEMAPHORES               200112L
#define _POSIX_SHARED_MEMORY_OBJECTS    200112L
#define _POSIX_SYNCHRONIZED_IO          200112L
#else
#define _POSIX_ASYNCHRONOUS_IO          1
#define _POSIX_FSYNC                    1
#define _POSIX_MAPPED_FILES             1
```

```
126 #define _POSIX_MEMLOCK                  1
127 #define _POSIX_MEMLOCK_RANGE            1
128 #define _POSIX_MEMORY_PROTECTION        1
129 #define _POSIX_MESSAGE_PASSING          1
130 #define _POSIX_PRIORITY_SCHEDULING      1
131 #define _POSIX_REALTIME_SIGNALS         1
132 #define _POSIX_SEMAPHORES               1
133 #define _POSIX_SHARED_MEMORY_OBJECTS    1
134 #define _POSIX_SYNCHRONIZED_IO          1
135 #endif

137 /*
138  * POSIX.1c compile-time symbolic constants.
139  */
140 #if defined(_XPG6)
141 #define _POSIX_THREAD_SAFE_FUNCTIONS            200112L
142 #define _POSIX_THREADS                          200112L
143 #define _POSIX_THREAD_ATTR_STACKADDR            200112L
144 #define _POSIX_THREAD_ATTR_STACKSIZE            200112L
145 #define _POSIX_THREAD_PROCESS_SHARED            200112L
146 #define _POSIX_THREAD_PRIORITY_SCHEDULING       200112L
147 #define _POSIX_TIMERS                           200112L
148 #else
149 #define _POSIX_THREAD_SAFE_FUNCTIONS            1
150 #define _POSIX_THREADS                          1
151 #define _POSIX_THREAD_ATTR_STACKADDR            1
152 #define _POSIX_THREAD_ATTR_STACKSIZE            1
153 #define _POSIX_THREAD_PROCESS_SHARED            1
154 #define _POSIX_THREAD_PRIORITY_SCHEDULING       1
155 #define _POSIX_TIMERS                           1
156 #endif

158 /* New in UNIX 03 */
159 #define _POSIX_ADVISORY_INFO                    200112L
160 #define _POSIX_BARRIERS                         200112L
161 #define _POSIX_CLOCK_SELECTION                  200112L
162 #define _POSIX_IPV6                             200112L
163 #define _POSIX_MONOTONIC_CLOCK                  200112L
164 #define _POSIX_RAW_SOCKETS                      200112L
165 #define _POSIX_READER_WRITER_LOCKS              200112L
166 #define _POSIX_SPAWN                            200112L
167 #define _POSIX_SPIN_LOCKS                       200112L
168 #define _POSIX_TIMEOUTS                         200112L

170 /*
171  * Support for the POSIX.1 mutex protocol attribute. For realtime applications
172  * which need mutexes to support priority inheritance/ceiling.
173  */
174 #if defined(_XPG6)
175 #define _POSIX_THREAD_PRIO_INHERIT              200112L
176 #define _POSIX_THREAD_PRIO_PROTECT              200112L
177 #else
178 #define _POSIX_THREAD_PRIO_INHERIT              1
179 #define _POSIX_THREAD_PRIO_PROTECT              1
180 #endif

182 #ifndef _POSIX_VDISABLE
183 #define _POSIX_VDISABLE         0
184 #endif

181 #ifndef NULL
182 #if defined(_LP64)
183 #define NULL    0L
184 #else
185 #define NULL    0
186 #endif
```



```
187 #endif

186 #define STDIN_FILENO    0
187 #define STDOUT_FILENO   1
188 #define STDERR_FILENO   2

190 /*
191  * Large File Summit-related announcement macros.  The system supports both
192  * the additional and transitional Large File Summit interfaces.  (The final
193  * two macros provide a finer granularity breakdown of _LFS64_LARGEFILE.)
194  */
195 #define _LFS_LARGEFILE          1
196 #define _LFS64_LARGEFILE        1
197 #define _LFS64_STDIO            1
198 #define _LFS64_ASYNCHRONOUS_IO  1

200 /* large file compilation environment setup */
201 #if !defined(_LP64) && _FILE_OFFSET_BITS == 64
202 #ifdef  __PRAGMA_REDEFINE_EXTNAME
203 #pragma redefine_extname        ftruncate       ftruncate64
204 #pragma redefine_extname        lseek           lseek64
205 #pragma redefine_extname        pread           pread64
206 #pragma redefine_extname        pwrite          pwrite64
207 #pragma redefine_extname        truncate        truncate64
208 #pragma redefine_extname        lockf           lockf64
209 #pragma redefine_extname        tell            tell64
210 #else   /* __PRAGMA_REDEFINE_EXTNAME */
211 #define ftruncate                       ftruncate64
212 #define lseek                           lseek64
213 #define pread                           pread64
214 #define pwrite                          pwrite64
215 #define truncate                        truncate64
216 #define lockf                           lockf64
217 #define tell                            tell64
218 #endif  /* __PRAGMA_REDEFINE_EXTNAME */
219 #endif  /* !_LP64 && _FILE_OFFSET_BITS == 64 */

221 /* In the LP64 compilation environment, the APIs are already large file */
222 #if defined(_LP64) && defined(_LARGEFILE64_SOURCE)
223 #ifdef  __PRAGMA_REDEFINE_EXTNAME
224 #pragma redefine_extname        ftruncate64     ftruncate
225 #pragma redefine_extname        lseek64         lseek
226 #pragma redefine_extname        pread64         pread
227 #pragma redefine_extname        pwrite64        pwrite
228 #pragma redefine_extname        truncate64      truncate
229 #pragma redefine_extname        lockf64         lockf
230 #pragma redefine_extname        tell64          tell
231 #else   /* __PRAGMA_REDEFINE_EXTNAME */
232 #define ftruncate64                     ftruncate
233 #define lseek64                         lseek
234 #define pread64                         pread
235 #define pwrite64                        pwrite
236 #define truncate64                      truncate
237 #define lockf64                         lockf
238 #define tell64                          tell
239 #endif  /* __PRAGMA_REDEFINE_EXTNAME */
240 #endif  /* _LP64 && _LARGEFILE64_SOURCE */

242 extern int access(const char *, int);
243 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
244 extern int acct(const char *);
245 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
246 extern unsigned alarm(unsigned);
247 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
248 #if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
249         defined(__EXTENSIONS__)
```

```
250 extern int brk(void *);
251 #endif /* !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2)... */
252 extern int chdir(const char *);
253 extern int chown(const char *, uid_t, gid_t);
254 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
255 #if !defined(_POSIX_C_SOURCE) || (defined(_XOPEN_SOURCE) && \
256         !defined(_XPG6)) || defined(__EXTENSIONS__)
257 extern int chroot(const char *);
258 #endif /* !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE))... */
259 extern int close(int);
260 #if defined(_XPG4) || defined(__EXTENSIONS__)
261 extern size_t confstr(int, char *, size_t);
262 extern char *crypt(const char *, const char *);
263 #endif /* defined(_XPG4) || defined(__EXTENSIONS__) */
264 #if !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE) || \
265         defined(__EXTENSIONS__)
266 extern char *ctermid(char *);
267 #endif /* (!defined(_POSIX_C_SOURCE) ... */
268 #if !defined(__XOPEN_OR_POSIX) || defined(_REENTRANT) || defined(__EXTENSIONS__)
269 extern char *ctermid_r(char *);
270 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_REENTRANT) ... */
271 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
272 #if !defined(_XPG6) || defined(__EXTENSIONS__)
273 extern char *cuserid(char *);
274 #endif
275 extern int dup(int);
276 extern int dup2(int, int);
277 extern int dup3(int, int, int);
278 #if defined(_XPG4) || defined(__EXTENSIONS__)
279 extern void encrypt(char *, int);
280 #endif /* defined(XPG4) || defined(__EXTENSIONS__) */
281 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
282 extern void endusershell(void);
283 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
284 extern int execl(const char *, const char *, ...);
285 extern int execle(const char *, const char *, ...);
286 extern int execlp(const char *, const char *, ...);
287 extern int execv(const char *, char *const *);
288 extern int execve(const char *, char *const *, char *const *);
289 extern int execvp(const char *, char *const *);
290 extern void _exit(int)
291         __NORETURN;
292 /*
293  * The following fattach prototype is duplicated in <stropts.h>. The
294  * duplication is necessitated by XPG4.2 which requires the prototype
295  * be defined in <stropts.h>.
296  */
297 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
298 extern int fattach(int, const char *);
299 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
300 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
301 extern int fchdir(int);
302 extern int fchown(int, uid_t, gid_t);
303 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
304 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
305 extern int fchroot(int);
306 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
307 #if !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2) || \
308         defined(__EXTENSIONS__)
309 extern int fdatasync(int);
310 #endif /* !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2)... */
311 /*
312  * The following fdetach prototype is duplicated in <stropts.h>. The
313  * duplication is necessitated by XPG4.2 which requires the prototype
314  * be defined in <stropts.h>.
315  */
```



```
316 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
317 extern int fdetach(const char *);
318 #endif /* !defined(__XOPEN_OR_POSIX)... */
319 extern pid_t fork(void);
320 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
321 extern pid_t fork1(void);
322 extern pid_t forkall(void);
323 #endif /* !defined(__XOPEN_OR_POSIX)... */
324 extern long fpathconf(int, int);
325 #if !defined(_POSIX_C_SOURCE) || (_POSIX_C_SOURCE > 2) || \
326         defined(__EXTENSIONS__)
327 extern int fsync(int);
328 #endif /* !defined(_POSIX_C_SOURCE) || (_POSIX_C_SOURCE > 2)... */
329 #if !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2) || defined(_XPG4_2) || \
330         (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
331         defined(__EXTENSIONS__)
332 extern int ftruncate(int, off_t);
333 #endif /* !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2)... */
334 extern char *getcwd(char *, size_t);
335 #if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
336         defined(__EXTENSIONS__)
337 extern int getdtablesize(void);
338 #endif
339 extern gid_t getegid(void);
340 extern uid_t geteuid(void);
341 extern gid_t getgid(void);
342 extern int getgroups(int, gid_t *);
343 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
344 extern long gethostid(void);
345 #endif
346 #if defined(_XPG4_2)
347 extern int gethostname(char *, size_t);
348 #elif  !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
349 extern int gethostname(char *, int);
350 #endif

352 #ifndef __GETLOGIN_DEFINED      /* Avoid duplicate in stdlib.h */
353 #define __GETLOGIN_DEFINED
354 #ifndef __USE_LEGACY_LOGNAME__
355 #ifdef  __PRAGMA_REDEFINE_EXTNAME
356 #pragma redefine_extname getlogin getloginx
357 #else   /* __PRAGMA_REDEFINE_EXTNAME */
358 extern char *getloginx(void);
359 #define getlogin        getloginx
360 #endif  /* __PRAGMA_REDEFINE_EXTNAME */
361 #endif  /* __USE_LEGACY_LOGNAME__ */
362 extern char *getlogin(void);
363 #endif  /* __GETLOGIN_DEFINED */

365 #if defined(_XPG4) || defined(__EXTENSIONS__)
366 extern int  getopt(int, char *const *, const char *);
367 extern char *optarg;
368 extern int  opterr, optind, optopt;
369 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
370 #if !defined(_XPG6) || defined(__EXTENSIONS__)
371 extern char *getpass(const char *);
372 #endif
373 #endif /* defined(_XPG4) || defined(__EXTENSIONS__) */
374 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
375 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
376 #if !defined(_XPG6) || defined(__EXTENSIONS__)
377 extern int getpagesize(void);
378 #endif
379 extern pid_t getpgid(pid_t);
380 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
381 extern pid_t getpid(void);
```

```
382 extern pid_t getppid(void);
383 extern pid_t getpgrp(void);

385 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
386 char *gettxt(const char *, const char *);
387 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
388 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
389 extern pid_t getsid(pid_t);
390 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
391 extern uid_t getuid(void);
392 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
393 extern char *getusershell(void);
394 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
395 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
396 extern char *getwd(char *);
397 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
398 /*
399  * The following ioctl prototype is duplicated in <stropts.h>. The
400  * duplication is necessitated by XPG4.2 which requires the prototype
401  * be defined in <stropts.h>.
402  */
403 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
404 extern int ioctl(int, int, ...);
405 extern int isaexec(const char *, char *const *, char *const *);
406 extern int issetugid(void);
407 #endif
408 extern int isatty(int);
409 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
410 extern int lchown(const char *, uid_t, gid_t);
411 #endif
412 extern int link(const char *, const char *);
413 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
414 extern offset_t llseek(int, offset_t, int);
415 #endif
416 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || \
417         (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
418         defined(__EXTENSIONS__)
419 extern int lockf(int, int, off_t);
420 #endif
421 extern off_t lseek(int, off_t, int);
422 #if !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE) || \
423         defined(__EXTENSIONS__)
424 extern int nice(int);
425 #endif /* !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)... */
426 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
427 extern int mincore(caddr_t, size_t, char *);
428 #endif
429 extern long pathconf(const char *, int);
430 extern int pause(void);
431 extern int pipe(int *);
432 extern int pipe2(int *, int);
433 #if !defined(_POSIX_C_SOURCE) || defined(_XPG5) || \
434         (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
435         defined(__EXTENSIONS__)
436 extern ssize_t pread(int, void *, size_t, off_t);
437 #endif
438 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
439 extern void profil(unsigned short *, size_t, unsigned long, unsigned int);
440 #endif
441 /*
442  * pthread_atfork() is also declared in <pthread.h> as per SUSv3. The
443  * declarations are identical. A change to either one may also require
444  * appropriate namespace updates in order to avoid redeclaration
445  * warnings in the case where both prototypes are exposed via inclusion
446  * of both <pthread.h> and <unistd.h>.
447  */
```



```
448 #if !defined(__XOPEN_OR_POSIX) || \
449         ((_POSIX_C_SOURCE > 2) && !defined(_XPG6)) || \
450         defined(__EXTENSIONS__)
451 extern int pthread_atfork(void (*) (void), void (*) (void), void (*) (void));
452 #endif /* !defined(__XOPEN_OR_POSIX) || ((_POSIX_C_SOURCE > 2) ... */
453 #if !defined(_LP64) && \
454         (!defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__))
455 extern int ptrace(int, pid_t, int, int);
456 #endif
457 #if !defined(_POSIX_C_SOURCE) || defined(_XPG5) || \
458         (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
459         defined(__EXTENSIONS__)
460 extern ssize_t pwrite(int, const void *, size_t, off_t);
461 #endif
462 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
463 /* per RFC 3542; This is also defined in netdb.h */
464 extern int rcmd_af(char **, unsigned short, const char *, const char *,
465         const char *, int *, int);
466 #endif
467 extern ssize_t read(int, void *, size_t);
468 #if !defined(__XOPEN_OR_POSIX) || \
469         defined(_XPG4_2) || defined(__EXTENSIONS__)
470 extern ssize_t readlink(const char *_RESTRICT_KYWD, char *_RESTRICT_KYWD,
471         size_t);
472 #endif
473 #if (!defined(__XOPEN_OR_POSIX) || (defined(_XPG3) && !defined(_XPG4))) || \
474         defined(__EXTENSIONS__)
475 #if __cplusplus >= 199711L
476 namespace std {
477 #endif
478 extern int rename(const char *, const char *);
479 #if __cplusplus >= 199711L
480 } /* end of namespace std */
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   44803 Sun Dec 14 23:31:23 2014
new/usr/src/lib/cfgadm_plugins/pci/common/cfga.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

/*ARGSUSED*/
cfga_err_t
cfga_private_func(const char *function, const char *ap_id,
    const char *options, struct cfga_confirm *confp,
    struct cfga_msg *msgp, char **errstring, cfga_flags_t flags)
{
        char *str;
        int   len, fd, i = 0, repeat = 0;
        char buf[MAXNAMELEN];
        char ptr;
        hpc_led_info_t  led_info;
        struct hpc_control_data iocdata;
        cfga_err_t rv;

        DBG(1, ("cfgadm_private_func: ap_id:%s\n", ap_id));
        DBG(2, ("  options: %s\n", (options == NULL)?"null":options));
        DBG(2, ("  confp: %x\n", confp));
        DBG(2, ("  cfga_msg: %x\n", cfga_msg));
        DBG(2, ("  flag: %d\n", flags));

        if ((rv = check_options(options)) != CFGA_OK) {
                return (rv);
        }

        if (private_check == confp)
                repeat = 1;
        else
                private_check = (void*)confp;

        /* XXX change const 6 to func_str[i] != NULL */
        for (i = 0, str = func_strs[i], len = strlen(str); i < 6; i++) {
                str = func_strs[i];
                len = strlen(str);
                if (strncmp(function, str, len) == 0)
                        break;
        }

        switch (i) {
                case ENABLE_SLOT:
                        build_control_data(&iocdata,
                                HPC_CTRL_ENABLE_SLOT, 0);
                        break;
                case DISABLE_SLOT:
                        build_control_data(&iocdata,
                                HPC_CTRL_DISABLE_SLOT, 0);
                        break;
                case ENABLE_AUTOCNF:
                        build_control_data(&iocdata,
                                HPC_CTRL_ENABLE_AUTOCFG, 0);
                        break;
                case DISABLE_AUTOCNF:
                        build_control_data(&iocdata,
                                HPC_CTRL_DISABLE_AUTOCFG, 0);
                        break;
                case LED:
                        /* set mode */
                        ptr = function[len++];
```



```
                        if (ptr == '=') {
                                str = (char *)function;
                                for (str = (str+len++), i = 0; *str != ',';
                                    i++, str++) {
                                        if (i == (MAXNAMELEN - 1))
                                                break;

                                        buf[i] = *str;
                                        DBG_F(2, (stdout, "%c\n", buf[i]));
                                }
                                buf[i] = '\0'; str++;
                                DBG(2, ("buf = %s\n", buf));

                                /* ACTIVE=3,ATTN=2,POWER=1,FAULT=0 */
                                if (strcmp(buf, led_strs[POWER]) == 0)
                                        led_info.led = HPC_POWER_LED;
                                else if (strcmp(buf, led_strs[FAULT]) == 0)
                                        led_info.led = HPC_FAULT_LED;
                                else if (strcmp(buf, led_strs[ATTN]) == 0)
                                        led_info.led = HPC_ATTN_LED;
                                else if (strcmp(buf, led_strs[ACTIVE]) == 0)
                                        led_info.led = HPC_ACTIVE_LED;
                                else return (CFGA_INVAL);

                                len = strlen(func_strs[MODE]);
                                if ((strncmp(str, func_strs[MODE], len) == 0) &&
                                    (*(str+(len)) == '=')) {
                                    for (str = (str+(++len)), i = 0;
                                        *str != '\0'; i++, str++) {
                                        *str != NULL; i++, str++) {
                                                buf[i] = *str;

                                    }
                                }
                                buf[i] = '\0';
                                DBG(2, ("buf_mode= %s\n", buf));

                                /* ON = 1, OFF = 0 */
                                if (strcmp(buf, mode_strs[ON]) == 0)
                                        led_info.state = HPC_LED_ON;
                                else if (strcmp(buf, mode_strs[OFF]) == 0)
                                        led_info.state = HPC_LED_OFF;
                                else if (strcmp(buf, mode_strs[BLINK]) == 0)
                                        led_info.state = HPC_LED_BLINK;
                                else return (CFGA_INVAL);

                                /* sendin  */
                                build_control_data(&iocdata,
                                    HPC_CTRL_SET_LED_STATE,
                                    (void *)&led_info);
                                break;
                        } else if (ptr == '\0') {
                                /* print mode */
                                DBG(1, ("Print mode\n"));
                                return (prt_led_mode(ap_id, repeat, errstring,
                                    msgp));
                        }
                default:
                        DBG(1, ("default\n"));
                        errno = EINVAL;
                        return (CFGA_INVAL);
        }

        if ((fd = open(ap_id, O_RDWR)) == -1) {
                DBG(1, ("open failed\n"));
                return (CFGA_ERROR);
```

```
1123         }

1125         DBG(1, ("open = ap_id=%s, fd=%d\n", ap_id, fd));

1127         if (ioctl(fd, DEVCTL_AP_CONTROL, &iocdata) == -1) {
1128                 DBG(1, ("ioctl failed\n"));
1129                 (void) close(fd);
1130                 return (CFGA_ERROR);
1131         }

1133         (void) close(fd);

1135         return (CFGA_OK);
1136 }
```

*unchanged_portion_omitted_*

```
1274 static int
1275 find_physical_slot_names(const char *devcomp, struct searcharg *slotarg)
1276 {
1277         di_node_t root_node;

1279         DBG(1, ("find_physical_slot_names\n"));

1281         if ((root_node = di_init("/", DINFOCPYALL|DINFOPATH))
1282                 == DI_NODE_NIL) {
1283                 DBG(1, ("di_init() failed\n"));
1284                 return (0);
1284                 return (NULL);
1285         }

1287         slotarg->devpath = (char *)devcomp;

1289         if ((slotarg->promp = di_prom_init()) == DI_PROM_HANDLE_NIL) {
1290                 DBG(1, ("di_prom_init() failed\n"));
1291                 di_fini(root_node);
1292                 return (0);
1292                 return (NULL);
1293         }

1295         (void) di_walk_minor(root_node, "ddi_ctl:attachment_point:pci",
1296                 0, (void *)slotarg, find_slotname);

1298         di_prom_fini(slotarg->promp);
1299         di_fini(root_node);
1300         if (slotarg->slotnames[0] != NULL)
1301                 return (0);
1302         else
1303                 return (-1);
1304 }
```

*unchanged_portion_omitted_*

```
**********************************************************
   39182 Sun Dec 14 23:31:23 2014
new/usr/src/lib/cfgadm_plugins/shp/common/shp.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_
```

```
 878 /*ARGSUSED*/
 879 cfga_err_t
 880 cfga_private_func(const char *function, const char *ap_id,
 881     const char *options, struct cfga_confirm *confp,
 882     struct cfga_msg *msgp, char **errstring, cfga_flags_t flags)
 883 {
 884         char *str;
 885         int   len, fd, i = 0, repeat = 0;
 886         char buf[MAXNAMELEN];
 887         char ptr;
 888         cfga_err_t rv;
 889         char *led, *mode;
 890         hp_node_t node;
 891         char *result;

 893         DBG(1, ("cfgadm_private_func: ap_id:%s\n", ap_id));
 894         DBG(2, ("  options: %s\n", (options == NULL)?"null":options));
 895         DBG(2, ("  confp: %x\n", confp));
 896         DBG(2, ("  cfga_msg: %x\n", cfga_msg));
 897         DBG(2, ("  flag: %d\n", flags));

 899         if ((rv = check_options(options)) != CFGA_OK) {
 900                 return (rv);
 901         }

 903         if (private_check == confp)
 904                 repeat = 1;
 905         else
 906                 private_check = (void*)confp;

 908         for (i = 0, str = func_strs[i], len = strlen(str);
 909             func_strs[i] != NULL; i++) {
 910                 str = func_strs[i];
 911                 len = strlen(str);
 912                 if (strncmp(function, str, len) == 0)
 913                         break;
 914         }

 916         switch (i) {
 917                 case ENABLE_SLOT:
 918                 case DISABLE_SLOT:
 919                         /* pass through */
 920                 case ENABLE_AUTOCNF:
 921                 case DISABLE_AUTOCNF:
 922                         /* no action needed */
 923                         return (CFGA_OK);
 924                         break;
 925                 case LED:
 926                         /* set mode */
 927                         ptr = function[len++];
 928                         if (ptr == '=') {
 929                                 str = (char *)function;
 930                                 for (str = (str+len++), i = 0; *str != ',';
 931                                     i++, str++) {
 932                                         if (i == (MAXNAMELEN - 1))
 933                                                 break;
```



```
 935                                         buf[i] = *str;
 936                                         DBG_F(2, (stdout, "%c\n", buf[i]));
 937                                 }
 938                                 buf[i] = '\0'; str++;
 939                                 DBG(2, ("buf = %s\n", buf));

 941                                 /* ACTIVE=3,ATTN=2,POWER=1,FAULT=0 */
 942                                 if (strcmp(buf, led_strs[POWER]) == 0)
 943                                         led = PCIEHPC_PROP_LED_POWER;
 944                                 else if (strcmp(buf, led_strs[FAULT]) == 0)
 945                                         led = PCIEHPC_PROP_LED_FAULT;
 946                                 else if (strcmp(buf, led_strs[ATTN]) == 0)
 947                                         led = PCIEHPC_PROP_LED_ATTN;
 948                                 else if (strcmp(buf, led_strs[ACTIVE]) == 0)
 949                                         led = PCIEHPC_PROP_LED_ACTIVE;
 950                                 else return (CFGA_INVAL);

 952                                 len = strlen(func_strs[MODE]);
 953                                 if ((strncmp(str, func_strs[MODE], len) == 0) &&
 954                                     (*(str+(len)) == '=')) {
 955                                         for (str = (str+(++len)), i = 0;
 956                                             *str != '\0'; i++, str++) {
 956                                             *str != NULL; i++, str++) {
 957                                                 buf[i] = *str;
 958                                         }
 959                                 }
 960                                 buf[i] = '\0';
 961                                 DBG(2, ("buf_mode= %s\n", buf));

 963                                 /* ON = 1, OFF = 0 */
 964                                 if (strcmp(buf, mode_strs[ON]) == 0)
 965                                         mode = PCIEHPC_PROP_VALUE_ON;
 966                                 else if (strcmp(buf, mode_strs[OFF]) == 0)
 967                                         mode = PCIEHPC_PROP_VALUE_OFF;
 968                                 else if (strcmp(buf, mode_strs[BLINK]) == 0)
 969                                         mode = PCIEHPC_PROP_VALUE_BLINK;
 970                                 else return (CFGA_INVAL);

 972                                 /* sendin  */
 973                                 memset(buf, 0, sizeof (buf));
 974                                 snprintf(buf, sizeof (buf), "%s=%s",
 975                                     led, mode);
 976                                 buf[MAXNAMELEN - 1] = '\0';

 978                                 break;
 979                         } else if (ptr == '\0') {
 980                                 /* print mode */
 981                                 DBG(1, ("Print mode\n"));
 982                                 return (prt_led_mode(ap_id, repeat, errstring,
 983                                     msgp));
 984                         }
 985                 default:
 986                         DBG(1, ("default\n"));
 987                         errno = EINVAL;
 988                         return (CFGA_INVAL);
 989         }

 991         rv = physpath2node(ap_id, errstring, &node);
 992         if (rv != CFGA_OK)
 993                 return (rv);

 995         if (hp_set_private(node, buf, &result) != 0) {
 996                 hp_fini(node);
 997                 return (CFGA_ERROR);
 998         }
```

```
1000         hp_fini(node);
1001         return (CFGA_OK);
1002 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   13676 Sun Dec 14 23:31:23 2014
new/usr/src/lib/fm/topo/modules/i86pc/chip/chip_intel.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 334 static int
 335 mc_nb_create(topo_mod_t *mod, uint16_t chip_smbid, tnode_t *pnode,
 336     const char *name, nvlist_t *auth, nvlist_t *nvl)
 337 {
 338         int err;
 339         int i, j;
 340         int channel;
 341         uint8_t nmc;
 342         uint8_t maxranks;
 343         uint8_t maxdimms;
 344         tnode_t *mcnode;
 345         nvlist_t *fmri;
 346         nvlist_t **channel_nvl;
 347         nvpair_t *nvp;
 348         char *pname;
 349         uint_t nchannels;

 351         if (nvlist_lookup_nvlist_array(nvl, MCINTEL_NVLIST_MC, &channel_nvl,
 352             &nchannels) != 0) {
 353                 whinge(mod, NULL,
 354                     "mc_nb_create: failed to find channel information\n");
 355                 return (-1);
 356         }
 357         if (nvlist_lookup_uint8(nvl, MCINTEL_NVLIST_NMEM, &nmc) == 0) {
 358                 /*
 359                  * Assume channels are evenly divided among the controllers.
 360                  * Convert nchannels to channels per controller
 361                  */
 362                 nchannels = nchannels / nmc;
 363         } else {
 364                 /*
 365                  * if number of memory controllers is not specified then there
 366                  * are two channels per controller and the nchannels is total
 367                  * we will set up nmc as number of controllers and convert
 368                  * nchannels to channels per controller
 369                  */
 370                 nmc = nchannels / 2;
 371                 nchannels = nchannels / nmc;
 372         }
 373         if (nvlist_lookup_uint8(nvl, MCINTEL_NVLIST_NRANKS, &maxranks) != 0)
 374                 maxranks = 2;
 375         if (nvlist_lookup_uint8(nvl, MCINTEL_NVLIST_NDIMMS, &maxdimms) != 0)
 376                 maxdimms = 0;
 377         if (topo_node_range_create(mod, pnode, name, 0, nmc-1) < 0) {
 378                 whinge(mod, NULL,
 379                     "mc_nb_create: node range create failed\n");
 380                 return (-1);
 381         }
 382         channel = 0;
 383         for (i = 0; i < nmc; i++) {
 384                 if (mkrsrc(mod, pnode, name, i, auth, &fmri) != 0) {
 385                         whinge(mod, NULL, "mc_nb_create: mkrsrc failed\n");
 386                         return (-1);
 387                 }
 388                 if ((mcnode = topo_node_bind(mod, pnode, name, i,
 389                     fmri)) == NULL) {
 390                         whinge(mod, NULL, "mc_nb_create: node bind failed"
```



```
 391                             " for memory-controller\n");
 392                         nvlist_free(fmri);
 393                         return (-1);
 394                 }

 396                 (void) topo_node_fru_set(mcnode, NULL, 0, &err);
 397                 nvlist_free(fmri);
 398                 (void) topo_pgroup_create(mcnode, &mc_pgroup, &err);

 400                 if (FM_AWARE_SMBIOS(mod))
 401                         (void) topo_node_label_set(mcnode, NULL, &err);

 403                 if (topo_node_range_create(mod, mcnode, DRAMCHANNEL, channel,
 404                     channel + nchannels - 1) < 0) {
 405                         whinge(mod, NULL,
 406                             "mc_nb_create: channel node range create failed\n");
 407                         return (-1);
 408                 }
 409                 for (j = 0; j < nchannels; j++) {
 410                         if (mc_add_channel(mod, chip_smbid, mcnode, channel,
 411                             auth, channel_nvl[channel], maxdimms,
 412                             maxranks) < 0) {
 413                                 return (-1);
 414                         }
 415                         channel++;
 416                 }
 417                 for (nvp = nvlist_next_nvpair(nvl, NULL); nvp != NULL;
 418                     nvp = nvlist_next_nvpair(nvl, nvp)) {
 419                         pname = nvpair_name(nvp);
 420                         if (strcmp(pname, MCINTEL_NVLIST_MC) != 0 &&
 421                             strcmp(pname, MCINTEL_NVLIST_NMEM) != 0 &&
 422                             strcmp(pname, MCINTEL_NVLIST_NRANKS) != 0 &&
 423                             strcmp(pname, MCINTEL_NVLIST_NDIMMS) != 0 &&
 424                             strcmp(pname, MCINTEL_NVLIST_VERSTR) != 0 &&
 425                             strcmp(pname, MCINTEL_NVLIST_MEM) != 0) {
 426                                 (void) nvprop_add(mod, nvp, PGNAME(MCT),
 427                                     mcnode);
 428                         }
 429                 }
 430         }

 432         return (0);
 432         return (NULL);
 433 }

 435 int
 436 mc_node_create(topo_mod_t *mod, uint16_t chip_smbid, tnode_t *pnode,
 437     const char *name, nvlist_t *auth)
 438 {
 439         mc_snapshot_info_t mcs;
 440         void *buf = NULL;
 441         nvlist_t *nvl;
 442         uint8_t ver;
 443         int rc;

 445         if (ioctl(mc_fd, MC_IOC_SNAPSHOT_INFO, &mcs) == -1 ||
 446             (buf = topo_mod_alloc(mod, mcs.mcs_size)) == NULL ||
 447             ioctl(mc_fd, MC_IOC_SNAPSHOT, buf) == -1) {

 449                 whinge(mod, NULL, "mc failed to snapshot %s\n",
 450                     strerror(errno));

 452                 free(buf);
 453                 (void) close(mc_fd);
 454                 return (0);
 454                 return (NULL);
```

```
455         }
456         (void) close(mc_fd);
457         (void) nvlist_unpack(buf, mcs.mcs_size, &nvl, 0);
458         topo_mod_free(mod, buf, mcs.mcs_size);

460         if (nvlist_lookup_uint8(nvl, MCINTEL_NVLIST_VERSTR, &ver) != 0) {
461                 whinge(mod, NULL, "mc nvlist is not versioned\n");
462                 nvlist_free(nvl);
463                 return (0);
463                 return (NULL);
464         } else if (ver != MCINTEL_NVLIST_VERS0) {
465                 whinge(mod, NULL, "mc nvlist version mismatch\n");
466                 nvlist_free(nvl);
467                 return (0);
467                 return (NULL);
468         }

470         rc = mc_nb_create(mod, chip_smbid, pnode, name, auth, nvl);

472         nvlist_free(nvl);
473         return (rc);
474 }
```

*unchanged_portion_omitted_*

****
15656 Sun Dec 14 23:31:24 2014
new/usr/src/lib/gss_mechs/mech_krb5/crypto/des/afsstring2key.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
****

_____unchanged_portion_omitted_

```
366
367
368 char *afs_crypt(const char *pw, const char *salt,
369                 /* must be at least 16 bytes */
370                 char *iobuf)
371 {
372         int i, j, c;
373         int temp;
374         char block[66];
375         char E[48];
376         /*
377          * The key schedule.
378          * Generated from the key.
379          */
380         char KS[16][48];
381
382         for(i=0; i<66; i++)
383                 block[i] = 0;
384         /* Solaris Kerberos */
385         for(i=0; ((c= *pw) != 0) && i<64; pw++){
385         for(i=0; ((c= *pw) != NULL) && i<64; pw++){
386                 for(j=0; j<7; j++, i++)
387                         block[i] = (c>>(6-j)) & 01;
388                 i++;
389         }
390
391         krb5_afs_crypt_setkey(block, E, KS);

393         for(i=0; i<66; i++)
394                 block[i] = 0;

396         for(i=0;i<2;i++){
397                 c = *salt++;
398                 iobuf[i] = c;
399                 if(c>'Z') c -= 6;
400                 if(c>'9') c -= 7;
401                 c -= '.';
402                 for(j=0;j<6;j++){
403                         if((c>>j) & 01){
404                                 temp = E[6*i+j];
405                                 E[6*i+j] = E[6*i+j+24];
406                                 E[6*i+j+24] = temp;
407                                 }
408                         }
409                 }
410
411         for(i=0; i<25; i++)
412                 krb5_afs_encrypt(block,E,KS);
413
414         for(i=0; i<11; i++){
415                 c = 0;
416                 for(j=0; j<6; j++){
417                         c <<= 1;
418                         c |= block[6*i+j];
419                         }
420                 c += '.';
421                 if(c>'9') c += 7;
422                 if(c>'Z') c += 6;
```



```
423                 iobuf[i+2] = c;
424         }
425         iobuf[i+2] = 0;
426         if(iobuf[1]==0)
427                 iobuf[1] = iobuf[0];
428         return(iobuf);
429 }
```

_____unchanged_portion_omitted_

```
**********************************************************
    7385 Sun Dec 14 23:31:24 2014
new/usr/src/lib/gss_mechs/mech_krb5/krb5/krb/vfy_increds.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
```

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

#include "k5-int.h"
#include "int-proto.h"

/* Solaris Kerberos */
extern krb5_error_code krb5_libdefault_boolean();

static krb5_error_code
krb5_cc_copy_creds_except(krb5_context context, krb5_ccache incc, krb5_ccache ou
{
   krb5_error_code code;
   krb5_flags flags;
   krb5_cc_cursor cur;
   krb5_creds creds;

   flags = 0;                           /* turns off OPENCLOSE mode */
   /* Solaris Kerberos */
   if ((code = krb5_cc_set_flags(context, incc, flags)) != 0)
   if ((code = krb5_cc_set_flags(context, incc, flags)) != NULL)
      return(code);
   /* Solaris Kerberos */
   if ((code = krb5_cc_set_flags(context, outcc, flags)) != 0)
   if ((code = krb5_cc_set_flags(context, outcc, flags)) != NULL)
      return(code);

   /* Solaris Kerberos */
   if ((code = krb5_cc_start_seq_get(context, incc, &cur)) != 0)
   if ((code = krb5_cc_start_seq_get(context, incc, &cur)) != NULL)
      goto cleanup;

   /* Solaris Kerberos */
   while ((code = krb5_cc_next_cred(context, incc, &cur, &creds)) == 0) {
   while ((code = krb5_cc_next_cred(context, incc, &cur, &creds)) == NULL) {
      if (krb5_principal_compare(context, princ, creds.server))
         continue;

      code = krb5_cc_store_cred(context, outcc, &creds);
      krb5_free_cred_contents(context, &creds);
      if (code)
         goto cleanup;
   }

   if (code != KRB5_CC_END)
      goto cleanup;

   code = 0;

cleanup:
   flags = KRB5_TC_OPENCLOSE;

   /* Solaris Kerberos */
   if (code)
      (void) krb5_cc_set_flags(context, incc, flags);
   else
      code = krb5_cc_set_flags(context, incc, flags);
```



```
   /* Solaris Kerberos */
   if (code)
      (void) krb5_cc_set_flags(context, outcc, flags);
   else
      code = krb5_cc_set_flags(context, outcc, flags);

   return(code);
}

krb5_error_code KRB5_CALLCONV
krb5_verify_init_creds(krb5_context context,
                       krb5_creds *creds,
                       krb5_principal server_arg,
                       krb5_keytab keytab_arg,
                       krb5_ccache *ccache_arg,
                       krb5_verify_init_creds_opt *options)
{
   krb5_error_code ret;
   krb5_principal server;
   krb5_keytab keytab;
   krb5_ccache ccache;
   krb5_keytab_entry kte;
   krb5_creds in_creds, *out_creds;
   krb5_auth_context authcon;
   krb5_data ap_req;

   /* KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN */

   server = NULL;
   keytab = NULL;
   ccache = NULL;
   out_creds = NULL;
   authcon = NULL;
   ap_req.data = NULL;

   /* Solaris Kerberos */
   if (server_arg)
      server = server_arg;
   else if (ret = krb5_sname_to_principal(context, NULL, NULL,
                                          KRB5_NT_SRV_HST, &server))
      goto cleanup;

   /* first, check if the server is in the keytab.  If not, there's
      no reason to continue.  rd_req does all this, but there's
      no way to know that a given error is caused by a missing
      keytab or key, and not by some other problem. */

   if (keytab_arg) {
      keytab = keytab_arg;
   } else {
       /* Solaris Kerberos: ignore errors here, deal with below */
      ret = krb5_kt_default(context, &keytab);
   }

   /*
    * Solaris Kerberos:
    * Warning: be very, very careful when modifying the logic here
    */
   if (keytab == NULL ||
       (ret = krb5_kt_get_entry(context, keytab, server, 0, 0, &kte))) {
       /* this means there is no keying material.  This is ok, as long as
          it is not prohibited by the configuration */
       /* Solaris Kerberos */
       int nofail = 1;  /* Solaris Kerberos: default return error if keytab prob
```

```
122         if (options &&
123             (options->flags & KRB5_VERIFY_INIT_CREDS_OPT_AP_REQ_NOFAIL)) {
124             /* first, if options are set then use the option value to set nofail
125              nofail = options->ap_req_nofail;
126         } else {
127             /*
128              * Solaris Kerberos:
129              * Check verify_ap_req_nofail if set in config file.  Note this logic
130              * assumes that krb5_libdefault_boolean will not set nofail to a
131              * default value if verify_ap_req_nofail is not explictly set in
132              * config file.  Don't care about the return code.
133              */
134             (void) krb5_libdefault_boolean(context, &creds->client->realm,
135                                            "verify_ap_req_nofail",
136                                            &nofail);
137         }
138         /* Solaris Kerberos: exit without an error ONLY if nofail is false */
139         if (!nofail)
140             ret = 0;
141
142         goto cleanup;
143     }
144
145     krb5_kt_free_entry(context, &kte);
146
147     /* If the creds are for the server principal, we're set, just do
148        a mk_req.  Otherwise, do a get_credentials first. */
149
150     if (krb5_principal_compare(context, server, creds->server)) {
151        /* make an ap_req */
152        if ((ret = krb5_mk_req_extended(context, &authcon, 0, NULL, creds,
153                                        &ap_req)))
154           goto cleanup;
155     } else {
156        /* this is unclean, but it's the easiest way without ripping the
157           library into very small pieces.  store the client's initial cred
158           in a memory ccache, then call the library.  Later, we'll copy
159           everything except the initial cred into the ccache we return to
160           the user.  A clean implementation would involve library
161           internals with a coherent idea of "in" and "out". */
162
163        /* insert the initial cred into the ccache */
164
165        if ((ret = krb5_cc_resolve(context, "MEMORY:rd_req", &ccache)))
166           goto cleanup;
167        /* Solaris Kerberos */
168        if ((ret = krb5_cc_initialize(context, ccache, creds->client)) != 0)
168        if ((ret = krb5_cc_initialize(context, ccache, creds->client)) != NULL)
169           goto cleanup;
170
171        /* Solaris Kerberos */
172        if ((ret = krb5_cc_store_cred(context, ccache, creds)) != 0)
172        if ((ret = krb5_cc_store_cred(context, ccache, creds)) != NULL)
173           goto cleanup;
174
175        /* set up for get_creds */
176        memset(&in_creds, 0, sizeof(in_creds));
177        in_creds.client = creds->client;
178        in_creds.server = server;
179        if ((ret = krb5_timeofday(context, &in_creds.times.endtime)))
180           goto cleanup;
181        in_creds.times.endtime += 5*60;
182
183        if ((ret = krb5_get_credentials(context, 0, ccache, &in_creds,
184                                        &out_creds)))
185           goto cleanup;
```



```
187        /* make an ap_req */
188        if ((ret = krb5_mk_req_extended(context, &authcon, 0, NULL, out_creds,
189                                        &ap_req)))
190           goto cleanup;
191     }
192
193     /* wipe the auth context for mk_req */
194     if (authcon) {
195        krb5_auth_con_free(context, authcon);
196        authcon = NULL;
197     }
198
199     /* verify the ap_req */
200
201     if ((ret = krb5_rd_req(context, &authcon, &ap_req, server, keytab,
202                            NULL, NULL)))
203        goto cleanup;
204
205     /* if we get this far, then the verification succeeded.  We can
206        still fail if the library stuff here fails, but that's it */
207
208     if (ccache_arg && ccache) {
209         if (*ccache_arg == NULL) {
210             krb5_ccache retcc;
211
212             retcc = NULL;
213
214             /* Solaris Kerberos */
215             if (((ret = krb5_cc_resolve(context, "MEMORY:rd_req2", &retcc)) != 0)
216                 ((ret = krb5_cc_initialize(context, retcc, creds->client)) != 0)
215             if (((ret = krb5_cc_resolve(context, "MEMORY:rd_req2", &retcc)) != NU
216                 ((ret = krb5_cc_initialize(context, retcc, creds->client)) != NUL
217                 ((ret = krb5_cc_copy_creds_except(context, ccache, retcc,
218                                                   creds->server)) != 0)) {
218                                                   creds->server)) != NULL)) {
219                 /* Solaris Kerberos */
220                 if (retcc)
221                     (void) krb5_cc_destroy(context, retcc);
222             } else {
223                 *ccache_arg = retcc;
224             }
225         } else {
226             ret = krb5_cc_copy_creds_except(context, ccache, *ccache_arg,
227                                             server);
228         }
229     }
230
231     /* if any of the above paths returned an errors, then ret is set
232        accordingly.  either that, or it's zero, which is fine, too */
233
234 cleanup:
235     if (!server_arg && server)
236        krb5_free_principal(context, server);
237      /* Solaris Kerberos */
238     if (!keytab_arg && keytab)
239        (void) krb5_kt_close(context, keytab);
240      /* Solaris Kerberos */
241     if (ccache)
242        (void) krb5_cc_destroy(context, ccache);
243     if (out_creds)
244        krb5_free_creds(context, out_creds);
245     if (authcon)
246        krb5_auth_con_free(context, authcon);
247     if (ap_req.data)
248        krb5_xfree(ap_req.data);
```

```
 250    return(ret);
 251 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    9589 Sun Dec 14 23:31:24 2014
new/usr/src/lib/libc/port/gen/getopt.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_

 175 /*
 176  * External function entry point.
 177  */
 178 int
 179 getopt(int argc, char *const *argv, const char *optstring)
 180 {
 181         char    c;
 182         char    *cp;
 183         int     longopt;
 184         char    *longoptarg;

 186         /*
 187          * Has the end of the options been encountered?  The following
 188          * implements the SUS requirements:
 189          *
 190          * If, when getopt() is called:
 191          *      argv[optind]    is a null pointer
 192          *      *argv[optind]   is not the character '-'
 193          *      argv[optind]    points to the string "-"
 194          * getopt() returns -1 without changing optind. If
 195          *      argv[optind]    points to the string "--"
 196          * getopt() returns -1 after incrementing optind.
 197          */
 198         if (_sp == 1) {
 199                 if (optind >= argc || argv[optind][0] != '-' ||
 200                     argv[optind] == NULL || argv[optind][1] == '\0')
 201                         return (EOF);
 202                 else if (strcmp(argv[optind], "--") == 0) {
 202                 else if (strcmp(argv[optind], "--") == NULL) {
 203                         optind++;
 204                         return (EOF);
 205                 }
 206         }

 208         /*
 209          * Getting this far indicates that an option has been encountered.
 210          * Note that the syntax of optstring applies special meanings to
 211          * the characters ':' and '(', so they are not permissible as
 212          * option letters. A special meaning is also applied to the ')'
 213          * character, but its meaning can be determined from context.
 214          * Note that the specification only requires that the alnum
 215          * characters be accepted.
 216          *
 217          * If the second character of the argument is a '-' this must be
 218          * a long-option, otherwise it must be a short option.  Scan for
 219          * the option in optstring by the appropriate algorithm. Either
 220          * scan will return a pointer to the short-option character in
 221          * optstring if the option is found and NULL otherwise.
 222          *
 223          * For an unrecognized long-option, optopt will equal 0, but
 224          * since long-options can't aggregate the failing option can
 225          * be identified by argv[optind-1].
 226          */
 227         optopt = c = (unsigned char)argv[optind][_sp];
 228         optarg = NULL;
 229         longopt = (_sp == 1 && c == '-');
 230         if (!(longopt ?
```



```
 231             ((cp = parselong(optstring, argv[optind]+2, &longoptarg)) != NULL) :
 232             ((cp = parseshort(optstring, c)) != NULL))) {
 233                 ERR(_libc_gettext("%s: illegal option -- %s\n"),
 234                     c, (longopt ? optind : 0));
 235                 /*
 236                  * Note: When the long option is unrecognized, optopt
 237                  * will be '-' here, which matches the specification.
 238                  */
 239                 if (argv[optind][++_sp] == '\0' || longopt) {
 240                         optind++;
 241                         _sp = 1;
 242                 }
 243                 return ('?');
 244         }
 245         optopt = c = *cp;

 247         /*
 248          * A valid option has been identified.  If it should have an
 249          * option-argument, process that now.  SUS defines the setting
 250          * of optarg as follows:
 251          *
 252          *   1. If the option was the last character in the string pointed to
 253          *      by an element of argv, then optarg contains the next element
 254          *      of argv, and optind is incremented by 2. If the resulting
 255          *      value of optind is not less than argc, this indicates a
 256          *      missing option-argument, and getopt() returns an error
 257          *      indication.
 258          *
 259          *   2. Otherwise, optarg points to the string following the option
 260          *      character in that element of argv, and optind is incremented
 261          *      by 1.
 262          *
 263          * The second clause allows -abcd (where b requires an option-argument)
 264          * to be interpreted as "-a -b cd".
 265          *
 266          * Note that the option-argument can legally be an empty string,
 267          * such as:
 268          *      command --option= operand
 269          * which explicitly sets the value of --option to nil
 270          */
 271         if (*(cp + 1) == ':') {
 272                 /* The option takes an argument */
 273                 if (!longopt && argv[optind][_sp+1] != '\0') {
 274                         optarg = &argv[optind++][_sp+1];
 275                 } else if (longopt && longoptarg) {
 276                         /*
 277                          * The option argument was explicitly set to
 278                          * the empty string on the command line (--option=)
 279                          */
 280                         optind++;
 281                         optarg = longoptarg;
 282                 } else if (++optind >= argc) {
 283                         ERR(_libc_gettext("%s: option requires an argument" \
 284                             " -- %s\n"), c, (longopt ? optind - 1 : 0));
 285                         _sp = 1;
 286                         optarg = NULL;
 287                         return (optstring[0] == ':' ? ':' : '?');
 288                 } else
 289                         optarg = argv[optind++];
 290                 _sp = 1;
 291         } else {
 292                 /* The option does NOT take an argument */
 293                 if (longopt && (longoptarg != NULL)) {
 294                         /* User supplied an arg to an option that takes none */
 295                         ERR(_libc_gettext(
 296                             "%s: option doesn't take an argument -- %s\n"),
```

```
                            0, (longopt ? optind : 0));
                        optarg = longoptarg = NULL;
                        c = '?';
                }

                if (longopt || argv[optind][++_sp] == '\0') {
                        _sp = 1;
                        optind++;
                }
                optarg = NULL;
        }
        return (c);
} /* getopt() */
```

unchanged_portion_omitted_

****
11835 Sun Dec 14 23:31:25 2014
new/usr/src/lib/libdtrace/common/dt_handle.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
****
_____***unchanged_portion_omitted_***

```
234 int
235 dt_handle_liberr(dtrace_hdl_t *dtp, const dtrace_probedata_t *data,
236     const char *faultstr)
237 {
238         dtrace_probedesc_t *errpd = data->dtpda_pdesc;
239         dtrace_errdata_t err;
240         const int slop = 80;
241         char *str;
242         int len;

244         err.dteda_edesc = data->dtpda_edesc;
245         err.dteda_pdesc = errpd;
246         err.dteda_cpu = data->dtpda_cpu;
247         err.dteda_action = -1;
248         err.dteda_offset = -1;
249         err.dteda_fault = DTRACEFLT_LIBRARY;
250         err.dteda_addr = 0L;
250         err.dteda_addr = NULL;

252         len = strlen(faultstr) +
253             strlen(errpd->dtpd_provider) + strlen(errpd->dtpd_mod) +
254             strlen(errpd->dtpd_name) + strlen(errpd->dtpd_func) +
255             slop;

257         str = alloca(len);

259         (void) snprintf(str, len, "error on enabled probe ID %u "
260             "(ID %u: %s:%s:%s:%s): %s\n",
261             data->dtpda_edesc->dtepd_epid,
262             errpd->dtpd_id, errpd->dtpd_provider,
263             errpd->dtpd_mod, errpd->dtpd_func,
264             errpd->dtpd_name, faultstr);

266         err.dteda_msg = str;

268         if (dtp->dt_errhdlr == NULL)
269                 return (dt_set_errno(dtp, EDT_ERRABORT));

271         if ((*dtp->dt_errhdlr)(&err, dtp->dt_errarg) == DTRACE_HANDLE_ABORT)
272                 return (dt_set_errno(dtp, EDT_ERRABORT));

274         return (0);
275 }
```

_____***unchanged_portion_omitted_***

****
73667 Sun Dec 14 23:31:25 2014
new/usr/src/lib/libscf/common/midlevel.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
****
_____***unchanged_portion_omitted_***

```
1302 /*
1303  * scf_general_pg_setup(fmri, pg_name)
1304  * Create a scf_simple_handle_t and fill in the instance, snapshot, and
1305  * property group fields associated with the given fmri and property group
1306  * name.
1307  * Returns:
1308  *      Handle  on success
1309  *      Null  on error with scf_error set to:
1310  *              SCF_ERROR_HANDLE_MISMATCH,
1311  *              SCF_ERROR_INVALID_ARGUMENT,
1312  *              SCF_ERROR_CONSTRAINT_VIOLATED,
1313  *              SCF_ERROR_NOT_FOUND,
1314  *              SCF_ERROR_NOT_SET,
1315  *              SCF_ERROR_DELETED,
1316  *              SCF_ERROR_NOT_BOUND,
1317  *              SCF_ERROR_CONNECTION_BROKEN,
1318  *              SCF_ERROR_INTERNAL,
1319  *              SCF_ERROR_NO_RESOURCES,
1320  *              SCF_ERROR_BACKEND_ACCESS
1321  */
1322 scf_simple_handle_t *
1323 scf_general_pg_setup(const char *fmri, const char *pg_name)
1324 {
1325         scf_simple_handle_t     *ret;

1327         ret = uu_zalloc(sizeof (*ret));
1328         if (ret == NULL) {
1329                 (void) scf_set_error(SCF_ERROR_NO_MEMORY);
1330                 return (NULL);
1331         } else {

1333                 ret->h = _scf_handle_create_and_bind(SCF_VERSION);
1334                 ret->inst = scf_instance_create(ret->h);
1335                 ret->snap = scf_snapshot_create(ret->h);
1336                 ret->running_pg = scf_pg_create(ret->h);
1337         }

1339         if ((ret->h == NULL) || (ret->inst == NULL) ||
1340             (ret->snap == NULL) || (ret->running_pg == NULL)) {
1341                 goto out;
1342         }

1344         if (scf_handle_decode_fmri(ret->h, fmri, NULL, NULL, ret->inst,
1345             NULL, NULL, 0) == -1) {
1345             NULL, NULL, NULL) == -1) {
1346                 goto out;
1347         }

1349         if ((scf_instance_get_snapshot(ret->inst, "running", ret->snap))
1350             != 0) {
1351                 goto out;
1352         }

1354         if (scf_instance_get_pg_composed(ret->inst, ret->snap, pg_name,
1355             ret->running_pg) != 0) {
1356                 goto out;
1357         }
```



```
1359         return (ret);

1361 out:
1362         scf_simple_handle_destroy(ret);
1363         return (NULL);
1364 }
```
_____***unchanged_portion_omitted_***

```
**********************************************************
    6854 Sun Dec 14 23:31:25 2014
new/usr/src/lib/libtnfctl/elf.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License, Version 1.0 only
 * (the "License").  You may not use this file except in compliance
 * with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 1994, by Sun Microsytems, Inc.
 */

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

/*
 * Interfaces for searching for elf specific information
 */

#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <link.h>
#include <sys/procfs.h>

#include "tnfctl_int.h"
#include "dbg.h"


/*
 * Declarations
 */

static tnfctl_errcode_t dynsec_num(tnfctl_handle_t *hndl, uintptr_t baseaddr,
                                   int objfd, int *num_dyn);
static tnfctl_errcode_t elf_dynmatch(Elf *elf, char *strs, Elf_Scn *dyn_scn,
        GElf_Shdr *dyn_shdr, Elf_Data *dyn_data,
        uintptr_t baseaddr, tnfctl_elf_search_t * search_info_p);
static tnfctl_errcode_t dyn_findtag(
        Elf3264_Dyn     *start,         /* start of dynam table read in */
        Elf3264_Sword   tag,            /* tag to search for */
        uintptr_t       dynam_addr,     /* address of _DYNAMIC in target */
        int             limit,          /* number of entries in table */
        uintptr_t       *dentry_address);       /* return value */
```



```
/* ------------------------------------------------------------------ */
/* ----------------------- Public Functions ------------------------- */
/* ------------------------------------------------------------------ */

/*
 * _tnfctl_elf_dbgent() - this function finds the address of the
 * debug struct (DT_DEBUG) in the target process.  _DYNAMIC is a symbol
 * present in every object.  The one in the main executable references
 * an array that is tagged with the kind of each member.  We search
 * for the tag of DT_DEBUG which is where the run time linker maintains
 * a structure that references the shared object linked list.
 *
 * A side effect of searching for DT_DEBUG ensures that the executable is
 * a dynamic executable - tracing only works on dynamic executables because
 * static executables don't have relocation tables.
 */
tnfctl_errcode_t
_tnfctl_elf_dbgent(tnfctl_handle_t *hndl, uintptr_t * entaddr_p)
{
        tnfctl_errcode_t        prexstat = TNFCTL_ERR_NONE;
        prb_status_t    prbstat = PRB_STATUS_OK;
        int             miscstat;
        int             objfd;
        int             num_dynentries = 0;
        uintptr_t       dynamic_addr;
        uintptr_t       baseaddr;
        uintptr_t       dentry_addr;
        Elf3264_Dyn     *dynam_tab = NULL;
        long            dynam_tab_size;

        *entaddr_p = (uintptr_t)NULL;
        *entaddr_p = NULL;

        prbstat = prb_mainobj_get(hndl->proc_p, &objfd, &baseaddr);
        if (prbstat)
                return (_tnfctl_map_to_errcode(prbstat));

        /* find the address of the symbol _DYNAMIC */
        prexstat = _tnfctl_sym_find_in_obj(objfd, baseaddr, "_DYNAMIC",
                        &dynamic_addr);
        if (prexstat) {
                prexstat = TNFCTL_ERR_NOTDYNAMIC;
                goto Cleanup;
        }

        /* find the number of entries in the .dynamic section */
        prexstat = dynsec_num(hndl, baseaddr, objfd, &num_dynentries);
        if (prexstat)
                goto Cleanup;

        DBG_TNF_PROBE_2(_tnfctl_elf_dbgent_1, "libtnfctl", "sunw%verbosity 2",
                tnf_long, num_of_dynentries, num_dynentries,
                tnf_opaque, DYNAMIC_address, dynamic_addr);

        /* read in the dynamic table from the image of the process */
        dynam_tab_size = num_dynentries * sizeof (Elf3264_Dyn);
        dynam_tab = malloc(dynam_tab_size);
        if (!dynam_tab) {
                close(objfd);
                return (TNFCTL_ERR_ALLOCFAIL);
        }
        miscstat = hndl->p_read(hndl->proc_p, dynamic_addr, dynam_tab,
                                                        dynam_tab_size);
        if (miscstat) {
                prexstat = TNFCTL_ERR_INTERNAL;
                goto Cleanup;
```

```
 125         }

 127         prexstat = dyn_findtag(dynam_tab, DT_DEBUG, dynamic_addr,
 128                 num_dynentries, &dentry_addr);
 129         if (prexstat) {
 130                 goto Cleanup;
 131         }
 132         *entaddr_p = dentry_addr;

 134 Cleanup:
 135         close(objfd);
 136         if (dynam_tab)
 137                 free(dynam_tab);
 138         return (prexstat);

 140 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
```
5162 Sun Dec 14 23:31:26 2014
new/usr/src/lib/libumem/common/misc.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
```
**********************************************************
```
_____***unchanged_portion_omitted_***

```
 238 /*
 239  * print_sym tries to print out the symbol and offset of a pointer
 240  */
 241 int
 242 print_sym(void *pointer)
 243 {
 244         int result;
 245         Dl_info sym_info;

 247         uintptr_t end = (uintptr_t)NULL;
 247         uintptr_t end = NULL;

 249         Sym *ext_info = NULL;

 251         result = dladdr1(pointer, &sym_info, (void **)&ext_info,
 252             RTLD_DL_SYMENT);

 254         if (result != 0) {
 255                 const char *endpath;

 257                 end = (uintptr_t)sym_info.dli_saddr + ext_info->st_size;

 259                 endpath = strrchr(sym_info.dli_fname, '/');
 260                 if (endpath)
 261                         endpath++;
 262                 else
 263                         endpath = sym_info.dli_fname;
 264                 umem_printf("%s'", endpath);
 265         }

 267         if (result == 0 || (uintptr_t)pointer > end) {
 268                 umem_printf("?? (0x%p)", pointer);
 269                 return (0);
 270         } else {
 271                 umem_printf("%s+0x%p", sym_info.dli_sname,
 272                     (char *)pointer - (char *)sym_info.dli_saddr);
 273                 return (1);
 274         }
 275 }
```
_____***unchanged_portion_omitted_***

```
**********************************************************
   90095 Sun Dec 14 23:31:26 2014
new/usr/src/pkg/manifests/system-header.mf
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
```

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

  22 #
  23 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
  24 # Copyright (c) 2012 by Delphix. All rights reserved.
  25 # Copyright 2012 Nexenta Systems, Inc. All rights reserved.
  26 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
  27 #

  29 set name=pkg.fmri value=pkg:/system/header@$(PKGVERS)
  30 set name=pkg.description \
  31     value="SunOS C/C++ header files for general development of software"
  32 set name=pkg.summary value="SunOS Header Files"
  33 set name=info.classification value=org.opensolaris.category.2008:System/Core
  34 set name=variant.arch value=$(ARCH)
  35 dir path=usr group=sys
  36 dir path=usr/include
  37 $(i386_ONLY)dir path=usr/include/$(ARCH64)
  38 $(i386_ONLY)dir path=usr/include/$(ARCH64)/sys
  39 dir path=usr/include/arpa
  40 dir path=usr/include/asm
  41 dir path=usr/include/ast
  42 dir path=usr/include/bsm
  43 dir path=usr/include/dat
  44 dir path=usr/include/des
  45 dir path=usr/include/gssapi
  46 dir path=usr/include/hal
  47 $(i386_ONLY)dir path=usr/include/ia32
  48 $(i386_ONLY)dir path=usr/include/ia32/sys
  49 dir path=usr/include/inet
  50 dir path=usr/include/inet/kssl
  51 dir path=usr/include/ipp
  52 dir path=usr/include/ipp/ipgpc
  53 dir path=usr/include/iso
  54 dir path=usr/include/kerberosv5
  55 dir path=usr/include/libpolkit
  56 dir path=usr/include/net
  57 dir path=usr/include/netinet
  58 dir path=usr/include/nfs
  59 dir path=usr/include/protocols
```



```
  60 dir path=usr/include/rpc
  61 dir path=usr/include/rpcsvc
  62 dir path=usr/include/sasl
  63 dir path=usr/include/scsi
  64 dir path=usr/include/scsi/plugins
  65 dir path=usr/include/scsi/plugins/ses
  66 dir path=usr/include/scsi/plugins/ses/framework
  67 dir path=usr/include/scsi/plugins/ses/vendor
  68 dir path=usr/include/scsi/plugins/smp
  69 dir path=usr/include/scsi/plugins/smp/engine
  70 dir path=usr/include/scsi/plugins/smp/framework
  71 dir path=usr/include/security
  72 dir path=usr/include/sharefs
  73 dir path=usr/include/sys
  74 dir path=usr/include/sys/av
  75 dir path=usr/include/sys/contract
  76 dir path=usr/include/sys/crypto
  77 dir path=usr/include/sys/dktp
  78 dir path=usr/include/sys/fc4
  79 dir path=usr/include/sys/fm
  80 dir path=usr/include/sys/fm/cpu
  81 dir path=usr/include/sys/fm/fs
  82 dir path=usr/include/sys/fm/io
  83 $(sparc_ONLY)dir path=usr/include/sys/fpu
  84 dir path=usr/include/sys/fs
  85 dir path=usr/include/sys/hotplug
  86 dir path=usr/include/sys/hotplug/pci
  87 dir path=usr/include/sys/ib
  88 dir path=usr/include/sys/ib/adapters
  89 dir path=usr/include/sys/ib/adapters/hermon
  90 dir path=usr/include/sys/ib/adapters/tavor
  91 dir path=usr/include/sys/ib/clients
  92 dir path=usr/include/sys/ib/clients/ibd
  93 dir path=usr/include/sys/ib/clients/of
  94 dir path=usr/include/sys/ib/clients/of/rdma
  95 dir path=usr/include/sys/ib/clients/of/sol_ofs
  96 dir path=usr/include/sys/ib/clients/of/sol_ucma
  97 dir path=usr/include/sys/ib/clients/of/sol_umad
  98 dir path=usr/include/sys/ib/clients/of/sol_uverbs
  99 dir path=usr/include/sys/ib/ibnex
 100 dir path=usr/include/sys/ib/ibtl
 101 dir path=usr/include/sys/ib/ibtl/impl
 102 dir path=usr/include/sys/ib/mgt
 103 dir path=usr/include/sys/ib/mgt/ibmf
 104 dir path=usr/include/sys/iso
 105 dir path=usr/include/sys/lvm
 106 dir path=usr/include/sys/proc
 107 dir path=usr/include/sys/rsm
 108 $(i386_ONLY)dir path=usr/include/sys/sata group=sys
 109 dir path=usr/include/sys/scsi
 110 dir path=usr/include/sys/scsi/adapters
 111 dir path=usr/include/sys/scsi/conf
 112 dir path=usr/include/sys/scsi/generic
 113 dir path=usr/include/sys/scsi/impl
 114 dir path=usr/include/sys/scsi/targets
 115 dir path=usr/include/sys/sysevent
 116 dir path=usr/include/sys/tsol
 117 dir path=usr/include/tsol
 118 dir path=usr/include/uuid
 119 $(sparc_ONLY)dir path=usr/include/v7
 120 $(sparc_ONLY)dir path=usr/include/v7/sys
 121 $(sparc_ONLY)dir path=usr/include/v9
 122 $(sparc_ONLY)dir path=usr/include/v9/sys
 123 dir path=usr/include/vm
 124 dir path=usr/platform group=sys
 125 $(sparc_ONLY)dir path=usr/platform/SUNW,A70 group=sys
```

```
$(sparc_ONLY)dir path=usr/platform/SUNW,Netra-CP2300 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Netra-CP2300/include
$(sparc_ONLY)dir path=usr/platform/SUNW,Netra-CP3010 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Netra-CP3010/include
$(sparc_ONLY)dir path=usr/platform/SUNW,Netra-T12 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Netra-T4 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,SPARC-Enterprise group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Serverblade1 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Blade-100 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Blade-1000 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Blade-1500 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Blade-2500 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-15000 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-280R group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-480R group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-880 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V215 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V240 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V250 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V440 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V445 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V490 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V890 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-2 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-250 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-4 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-Enterprise group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-Enterprise-10000 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,UltraSPARC-IIe-NetraCT-40 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,UltraSPARC-IIe-NetraCT-60 group=sys
$(sparc_ONLY)dir path=usr/platform/SUNW,UltraSPARC-IIi-Netract group=sys
$(i386_ONLY)dir path=usr/platform/i86pc group=sys
$(i386_ONLY)dir path=usr/platform/i86pc/include
$(i386_ONLY)dir path=usr/platform/i86pc/include/sys
$(i386_ONLY)dir path=usr/platform/i86pc/include/vm
$(i386_ONLY)dir path=usr/platform/i86xpv group=sys
$(i386_ONLY)dir path=usr/platform/i86xpv/include
$(i386_ONLY)dir path=usr/platform/i86xpv/include/sys
$(i386_ONLY)dir path=usr/platform/i86xpv/include/vm
$(sparc_ONLY)dir path=usr/platform/sun4u group=sys
$(sparc_ONLY)dir path=usr/platform/sun4u/include
$(sparc_ONLY)dir path=usr/platform/sun4u/include/sys
$(sparc_ONLY)dir path=usr/platform/sun4u/include/sys/i2c
$(sparc_ONLY)dir path=usr/platform/sun4u/include/sys/i2c/clients
$(sparc_ONLY)dir path=usr/platform/sun4u/include/sys/i2c/misc
$(sparc_ONLY)dir path=usr/platform/sun4u/include/vm
$(sparc_ONLY)dir path=usr/platform/sun4v group=sys
$(sparc_ONLY)dir path=usr/platform/sun4v/include
$(sparc_ONLY)dir path=usr/platform/sun4v/include/sys
$(sparc_ONLY)dir path=usr/platform/sun4v/include/vm
dir path=usr/share
dir path=usr/share/man
dir path=usr/share/man/man3head
dir path=usr/share/man/man4
dir path=usr/share/man/man5
dir path=usr/share/man/man7i
dir path=usr/share/src group=sys
dir path=usr/share/src/uts
$(i386_ONLY)dir path=usr/share/src/uts/i86pc
$(i386_ONLY)dir path=usr/share/src/uts/i86xpv
$(sparc_ONLY)dir path=usr/share/src/uts/sun4u
$(sparc_ONLY)dir path=usr/share/src/uts/sun4v
dir path=usr/xpg4
dir path=usr/xpg4/include
$(i386_ONLY)file path=usr/include/$(ARCH64)/sys/kdi_regs.h
```



```
$(i386_ONLY)file path=usr/include/$(ARCH64)/sys/privmregs.h
$(i386_ONLY)file path=usr/include/$(ARCH64)/sys/privregs.h
file path=usr/include/aio.h
file path=usr/include/alloca.h
file path=usr/include/apptrace.h
file path=usr/include/apptrace_impl.h
file path=usr/include/ar.h
file path=usr/include/archives.h
file path=usr/include/arpa/ftp.h
file path=usr/include/arpa/inet.h
file path=usr/include/arpa/nameser.h
file path=usr/include/arpa/nameser_compat.h
file path=usr/include/arpa/telnet.h
file path=usr/include/arpa/tftp.h
$(i386_ONLY)file path=usr/include/asm/atomic.h
$(i386_ONLY)file path=usr/include/asm/bitmap.h
$(i386_ONLY)file path=usr/include/asm/byteorder.h
$(i386_ONLY)file path=usr/include/asm/clock.h
$(i386_ONLY)file path=usr/include/asm/cpu.h
$(i386_ONLY)file path=usr/include/asm/cpuvar.h
$(sparc_ONLY)file path=usr/include/asm/flush.h
$(i386_ONLY)file path=usr/include/asm/htable.h
$(i386_ONLY)file path=usr/include/asm/mmu.h
file path=usr/include/asm/sunddi.h
file path=usr/include/asm/thread.h
file path=usr/include/assert.h
file path=usr/include/ast/align.h
file path=usr/include/ast/ast.h
file path=usr/include/ast/ast_botch.h
file path=usr/include/ast/ast_ccode.h
file path=usr/include/ast/ast_common.h
file path=usr/include/ast/ast_dir.h
file path=usr/include/ast/ast_dirent.h
file path=usr/include/ast/ast_fcntl.h
file path=usr/include/ast/ast_float.h
file path=usr/include/ast/ast_fs.h
file path=usr/include/ast/ast_getopt.h
file path=usr/include/ast/ast_iconv.h
file path=usr/include/ast/ast_lib.h
file path=usr/include/ast/ast_limits.h
file path=usr/include/ast/ast_map.h
file path=usr/include/ast/ast_mmap.h
file path=usr/include/ast/ast_mode.h
file path=usr/include/ast/ast_namval.h
file path=usr/include/ast/ast_ndbm.h
file path=usr/include/ast/ast_nl_types.h
file path=usr/include/ast/ast_param.h
file path=usr/include/ast/ast_standards.h
file path=usr/include/ast/ast_std.h
file path=usr/include/ast/ast_stdio.h
file path=usr/include/ast/ast_sys.h
file path=usr/include/ast/ast_time.h
file path=usr/include/ast/ast_tty.h
file path=usr/include/ast/ast_version.h
file path=usr/include/ast/ast_vfork.h
file path=usr/include/ast/ast_wait.h
file path=usr/include/ast/ast_wchar.h
file path=usr/include/ast/ast_windows.h
file path=usr/include/ast/bytesex.h
file path=usr/include/ast/ccode.h
file path=usr/include/ast/cdt.h
file path=usr/include/ast/cmd.h
file path=usr/include/ast/cmdext.h
file path=usr/include/ast/debug.h
file path=usr/include/ast/dirent.h
file path=usr/include/ast/dlldefs.h
```

```
258 file path=usr/include/ast/dt.h
259 file path=usr/include/ast/endian.h
260 file path=usr/include/ast/error.h
261 file path=usr/include/ast/find.h
262 file path=usr/include/ast/fnmatch.h
263 file path=usr/include/ast/fnv.h
264 file path=usr/include/ast/fs3d.h
265 file path=usr/include/ast/fts.h
266 file path=usr/include/ast/ftw.h
267 file path=usr/include/ast/ftwalk.h
268 file path=usr/include/ast/getopt.h
269 file path=usr/include/ast/glob.h
270 file path=usr/include/ast/hash.h
271 file path=usr/include/ast/hashkey.h
272 file path=usr/include/ast/hashpart.h
273 file path=usr/include/ast/history.h
274 file path=usr/include/ast/iconv.h
275 file path=usr/include/ast/ip6.h
276 file path=usr/include/ast/lc.h
277 file path=usr/include/ast/ls.h
278 file path=usr/include/ast/magic.h
279 file path=usr/include/ast/magicid.h
280 file path=usr/include/ast/mc.h
281 file path=usr/include/ast/mime.h
282 file path=usr/include/ast/mnt.h
283 file path=usr/include/ast/modecanon.h
284 file path=usr/include/ast/modex.h
285 file path=usr/include/ast/namval.h
286 file path=usr/include/ast/nl_types.h
287 file path=usr/include/ast/nval.h
288 file path=usr/include/ast/option.h
289 file path=usr/include/ast/preroot.h
290 file path=usr/include/ast/proc.h
291 file path=usr/include/ast/prototyped.h
292 file path=usr/include/ast/re_comp.h
293 file path=usr/include/ast/recfmt.h
294 file path=usr/include/ast/regex.h
295 file path=usr/include/ast/regexp.h
296 file path=usr/include/ast/sfdisc.h
297 file path=usr/include/ast/sfio.h
298 file path=usr/include/ast/sfio_s.h
299 file path=usr/include/ast/sfio_t.h
300 file path=usr/include/ast/shcmd.h
301 file path=usr/include/ast/shell.h
302 file path=usr/include/ast/sig.h
303 file path=usr/include/ast/stack.h
304 file path=usr/include/ast/stak.h
305 file path=usr/include/ast/stdio.h
306 file path=usr/include/ast/stk.h
307 file path=usr/include/ast/sum.h
308 file path=usr/include/ast/swap.h
309 file path=usr/include/ast/tar.h
310 file path=usr/include/ast/times.h
311 file path=usr/include/ast/tm.h
312 file path=usr/include/ast/tmx.h
313 file path=usr/include/ast/tok.h
314 file path=usr/include/ast/tv.h
315 file path=usr/include/ast/usage.h
316 file path=usr/include/ast/vdb.h
317 file path=usr/include/ast/vecargs.h
318 file path=usr/include/ast/vmalloc.h
319 file path=usr/include/ast/wait.h
320 file path=usr/include/ast/wchar.h
321 file path=usr/include/ast/wordexp.h
322 file path=usr/include/atomic.h
323 file path=usr/include/attr.h
```



```
324 file path=usr/include/auth_attr.h
325 file path=usr/include/bsm/adt.h
326 file path=usr/include/bsm/adt_event.h
327 file path=usr/include/bsm/audit.h
328 file path=usr/include/bsm/audit_kernel.h
329 file path=usr/include/bsm/audit_kevents.h
330 file path=usr/include/bsm/audit_record.h
331 file path=usr/include/bsm/audit_uevents.h
332 file path=usr/include/bsm/devices.h
333 file path=usr/include/bsm/libbsm.h
334 file path=usr/include/config_admin.h
335 file path=usr/include/cpio.h
336 file path=usr/include/crypt.h
337 file path=usr/include/cryptoutil.h
338 file path=usr/include/ctype.h
339 file path=usr/include/curses.h
340 file path=usr/include/dat/dat.h
341 file path=usr/include/dat/dat_error.h
342 file path=usr/include/dat/dat_platform_specific.h
343 file path=usr/include/dat/dat_redirection.h
344 file path=usr/include/dat/dat_registry.h
345 file path=usr/include/dat/dat_vendor_specific.h
346 file path=usr/include/dat/udat.h
347 file path=usr/include/dat/udat_config.h
348 file path=usr/include/dat/udat_redirection.h
349 file path=usr/include/dat/udat_vendor_specific.h
350 file path=usr/include/deflt.h
351 file path=usr/include/des/des.h
352 file path=usr/include/des/desdata.h
353 file path=usr/include/des/softdes.h
354 file path=usr/include/device_info.h
355 file path=usr/include/devid.h
356 file path=usr/include/devmgmt.h
357 file path=usr/include/devpoll.h
358 file path=usr/include/dial.h
359 file path=usr/include/dirent.h
360 file path=usr/include/dlfcn.h
361 file path=usr/include/door.h
362 file path=usr/include/elf.h
363 file path=usr/include/err.h
364 file path=usr/include/errno.h
365 file path=usr/include/eti.h
366 file path=usr/include/euc.h
367 file path=usr/include/exacct.h
368 file path=usr/include/exacct_impl.h
369 file path=usr/include/exec_attr.h
370 file path=usr/include/execinfo.h
371 file path=usr/include/fatal.h
372 file path=usr/include/fcntl.h
373 file path=usr/include/float.h
374 file path=usr/include/fmtmsg.h
375 file path=usr/include/fnmatch.h
376 file path=usr/include/form.h
377 file path=usr/include/ftw.h
378 file path=usr/include/gelf.h
379 file path=usr/include/getopt.h
380 file path=usr/include/getwidth.h
381 file path=usr/include/glob.h
382 file path=usr/include/grp.h
383 file path=usr/include/gssapi/gssapi.h
384 file path=usr/include/gssapi/gssapi_ext.h
385 file path=usr/include/hal/libhal-storage.h
386 file path=usr/include/hal/libhal.h
387 $(i386_ONLY)file path=usr/include/ia32/sys/asm_linkage.h
388 $(i386_ONLY)file path=usr/include/ia32/sys/kdi_regs.h
389 $(i386_ONLY)file path=usr/include/ia32/sys/machtypes.h
```

```
390 $(i386_ONLY)file path=usr/include/ia32/sys/privmregs.h
391 $(i386_ONLY)file path=usr/include/ia32/sys/privregs.h
392 $(i386_ONLY)file path=usr/include/ia32/sys/psw.h
393 $(i386_ONLY)file path=usr/include/ia32/sys/pte.h
394 $(i386_ONLY)file path=usr/include/ia32/sys/reg.h
395 $(i386_ONLY)file path=usr/include/ia32/sys/stack.h
396 $(i386_ONLY)file path=usr/include/ia32/sys/trap.h
397 $(i386_ONLY)file path=usr/include/ia32/sys/traptrace.h
398 file path=usr/include/iconv.h
399 file path=usr/include/idmap.h
400 file path=usr/include/ieeefp.h
401 file path=usr/include/ifaddrs.h
402 file path=usr/include/inet/arp.h
403 file path=usr/include/inet/common.h
404 file path=usr/include/inet/ip.h
405 file path=usr/include/inet/ip6.h
406 file path=usr/include/inet/ip6_asp.h
407 file path=usr/include/inet/ip_arp.h
408 file path=usr/include/inet/ip_ftable.h
409 file path=usr/include/inet/ip_if.h
410 file path=usr/include/inet/ip_ire.h
411 file path=usr/include/inet/ip_multi.h
412 file path=usr/include/inet/ip_netinfo.h
413 file path=usr/include/inet/ip_rts.h
414 file path=usr/include/inet/ip_stack.h
415 file path=usr/include/inet/ipclassifier.h
416 file path=usr/include/inet/ipdrop.h
417 file path=usr/include/inet/ipnet.h
418 file path=usr/include/inet/ipp_common.h
419 file path=usr/include/inet/kssl/ksslapi.h
420 file path=usr/include/inet/led.h
421 file path=usr/include/inet/mi.h
422 file path=usr/include/inet/mib2.h
423 file path=usr/include/inet/nd.h
424 file path=usr/include/inet/optcom.h
425 file path=usr/include/inet/sctp_itf.h
426 file path=usr/include/inet/snmpcom.h
427 file path=usr/include/inet/tcp.h
428 file path=usr/include/inet/tcp_sack.h
429 file path=usr/include/inet/tcp_stack.h
430 file path=usr/include/inet/tcp_stats.h
431 file path=usr/include/inet/tunables.h
432 file path=usr/include/inet/wifi_ioctl.h
433 file path=usr/include/inttypes.h
434 file path=usr/include/ipmp.h
435 file path=usr/include/ipmp_admin.h
436 file path=usr/include/ipmp_mpathd.h
437 file path=usr/include/ipmp_query.h
438 file path=usr/include/ipp/ipgpc/ipgpc.h
439 file path=usr/include/ipp/ipp.h
440 file path=usr/include/ipp/ipp_config.h
441 file path=usr/include/ipp/ipp_impl.h
442 file path=usr/include/ipp/ippctl.h
443 file path=usr/include/iso/ctype_iso.h
444 file path=usr/include/iso/limits_iso.h
445 file path=usr/include/iso/locale_iso.h
446 file path=usr/include/iso/setjmp_iso.h
447 file path=usr/include/iso/signal_iso.h
448 file path=usr/include/iso/stdarg_c99.h
449 file path=usr/include/iso/stdarg_iso.h
450 file path=usr/include/iso/stddef_iso.h
451 file path=usr/include/iso/stdio_c99.h
452 file path=usr/include/iso/stdio_iso.h
453 file path=usr/include/iso/stdlib_c99.h
454 file path=usr/include/iso/stdlib_iso.h
455 file path=usr/include/iso/string_iso.h
```



```
456 file path=usr/include/iso/time_iso.h
457 file path=usr/include/iso/wchar_c99.h
458 file path=usr/include/iso/wchar_iso.h
459 file path=usr/include/iso/wctype_iso.h
460 file path=usr/include/iso646.h
461 file path=usr/include/kerberosv5/com_err.h
462 file path=usr/include/kerberosv5/krb5.h
463 file path=usr/include/kerberosv5/mit-sipb-copyright.h
464 file path=usr/include/kerberosv5/mit_copyright.h
465 file path=usr/include/klpd.h
466 file path=usr/include/kmfapi.h
467 file path=usr/include/kmftypes.h
468 file path=usr/include/kstat.h
469 file path=usr/include/kvm.h
470 file path=usr/include/langinfo.h
471 file path=usr/include/lastlog.h
472 file path=usr/include/lber.h
473 file path=usr/include/ldap.h
474 file path=usr/include/libcontract.h
475 file path=usr/include/libctf.h
476 file path=usr/include/libdevice.h
477 file path=usr/include/libdevinfo.h
478 file path=usr/include/libdladm.h
479 file path=usr/include/libdlbridge.h
480 file path=usr/include/libdlib.h
481 file path=usr/include/libdllink.h
482 file path=usr/include/libdlpi.h
483 file path=usr/include/libdlvlan.h
484 file path=usr/include/libelf.h
485 $(i386_ONLY)file path=usr/include/libfdisk.h
486 file path=usr/include/libfstyp.h
487 file path=usr/include/libfstyp_module.h
488 file path=usr/include/libgen.h
489 file path=usr/include/libgrubmgmt.h
490 file path=usr/include/libintl.h
491 file path=usr/include/libipmi.h
492 file path=usr/include/libipp.h
493 file path=usr/include/libnvpair.h
494 file path=usr/include/libnwam.h
495 file path=usr/include/libpolkit/libpolkit.h
496 file path=usr/include/librcm.h
497 file path=usr/include/libscf.h
498 file path=usr/include/libscf_priv.h
499 file path=usr/include/libshare.h
500 file path=usr/include/libsvm.h
501 file path=usr/include/libsysevent.h
502 file path=usr/include/libsysevent_impl.h
503 file path=usr/include/libtsnet.h
504 $(sparc_ONLY)file path=usr/include/libv12n.h
505 file path=usr/include/libw.h
506 file path=usr/include/libzfs.h
507 file path=usr/include/libzfs_core.h
508 file path=usr/include/libzoneinfo.h
509 file path=usr/include/limits.h
510 file path=usr/include/linenum.h
511 file path=usr/include/link.h
512 file path=usr/include/listen.h
513 file path=usr/include/locale.h
514 file path=usr/include/macros.h
515 file path=usr/include/maillock.h
516 file path=usr/include/malloc.h
517 file path=usr/include/md4.h
518 file path=usr/include/md5.h
519 file path=usr/include/mdiox.h
520 file path=usr/include/mdmn_changelog.h
521 file path=usr/include/memory.h
```

```
522 file path=usr/include/menu.h
523 file path=usr/include/meta.h
524 file path=usr/include/meta_basic.h
525 file path=usr/include/meta_runtime.h
526 file path=usr/include/metacl.h
527 file path=usr/include/metad.h
528 file path=usr/include/metadyn.h
529 file path=usr/include/metamed.h
530 file path=usr/include/metamhd.h
531 file path=usr/include/mhdx.h
532 file path=usr/include/mon.h
533 file path=usr/include/monetary.h
534 file path=usr/include/mp.h
535 file path=usr/include/mqueue.h
536 file path=usr/include/mtmalloc.h
537 file path=usr/include/nan.h
538 file path=usr/include/ndbm.h
539 file path=usr/include/ndpd.h
540 file path=usr/include/net/af.h
541 file path=usr/include/net/bridge.h
542 file path=usr/include/net/if.h
543 file path=usr/include/net/if_arp.h
544 file path=usr/include/net/if_dl.h
545 file path=usr/include/net/if_types.h
546 file path=usr/include/net/pfkeyv2.h
547 file path=usr/include/net/pfpolicy.h
548 file path=usr/include/net/ppp-comp.h
549 file path=usr/include/net/ppp_defs.h
550 file path=usr/include/net/pppio.h
551 file path=usr/include/net/radix.h
552 file path=usr/include/net/route.h
553 file path=usr/include/net/trill.h
554 file path=usr/include/net/vjcompress.h
555 file path=usr/include/netconfig.h
556 file path=usr/include/netdb.h
557 file path=usr/include/netdir.h
558 file path=usr/include/netinet/arp.h
559 file path=usr/include/netinet/dhcp.h
560 file path=usr/include/netinet/dhcp6.h
561 file path=usr/include/netinet/icmp6.h
562 file path=usr/include/netinet/icmp_var.h
563 file path=usr/include/netinet/if_ether.h
564 file path=usr/include/netinet/igmp.h
565 file path=usr/include/netinet/igmp_var.h
566 file path=usr/include/netinet/in.h
567 file path=usr/include/netinet/in_pcb.h
568 file path=usr/include/netinet/in_systm.h
569 file path=usr/include/netinet/in_var.h
570 file path=usr/include/netinet/ip.h
571 file path=usr/include/netinet/ip6.h
572 file path=usr/include/netinet/ip_icmp.h
573 file path=usr/include/netinet/ip_mroute.h
574 file path=usr/include/netinet/ip_var.h
575 file path=usr/include/netinet/pim.h
576 file path=usr/include/netinet/sctp.h
577 file path=usr/include/netinet/tcp.h
578 file path=usr/include/netinet/tcp_debug.h
579 file path=usr/include/netinet/tcp_fsm.h
580 file path=usr/include/netinet/tcp_seq.h
581 file path=usr/include/netinet/tcp_timer.h
582 file path=usr/include/netinet/tcp_var.h
583 file path=usr/include/netinet/tcpip.h
584 file path=usr/include/netinet/udp.h
585 file path=usr/include/netinet/udp_var.h
586 file path=usr/include/netinet/vrrp.h
587 file path=usr/include/nfs/auth.h
```



```
588 file path=usr/include/nfs/export.h
589 file path=usr/include/nfs/lm.h
590 file path=usr/include/nfs/mapid.h
591 file path=usr/include/nfs/mount.h
592 file path=usr/include/nfs/nfs.h
593 file path=usr/include/nfs/nfs4.h
594 file path=usr/include/nfs/nfs4_attr.h
595 file path=usr/include/nfs/nfs4_clnt.h
596 file path=usr/include/nfs/nfs4_db_impl.h
597 file path=usr/include/nfs/nfs4_idmap_impl.h
598 file path=usr/include/nfs/nfs4_kprot.h
599 file path=usr/include/nfs/nfs_acl.h
600 file path=usr/include/nfs/nfs_clnt.h
601 file path=usr/include/nfs/nfs_cmd.h
602 file path=usr/include/nfs/nfs_log.h
603 file path=usr/include/nfs/nfs_sec.h
604 file path=usr/include/nfs/nfsid_map.h
605 file path=usr/include/nfs/nfssys.h
606 file path=usr/include/nfs/rnode.h
607 file path=usr/include/nfs/rnode4.h
608 file path=usr/include/nl_types.h
609 file path=usr/include/nlist.h
610 file path=usr/include/note.h
611 file path=usr/include/nss_common.h
612 file path=usr/include/nss_dbdefs.h
613 file path=usr/include/nss_netdir.h
614 file path=usr/include/nsswitch.h
615 file path=usr/include/panel.h
616 file path=usr/include/paths.h
617 file path=usr/include/pcsample.h
618 file path=usr/include/pfmt.h
619 file path=usr/include/pkgdev.h
620 file path=usr/include/pkginfo.h
621 file path=usr/include/pkglocs.h
622 file path=usr/include/pkgstrct.h
623 file path=usr/include/pkgtrans.h
624 file path=usr/include/poll.h
625 file path=usr/include/port.h
626 file path=usr/include/priv.h
627 file path=usr/include/proc_service.h
628 file path=usr/include/procfs.h
629 file path=usr/include/prof.h
630 file path=usr/include/prof_attr.h
631 file path=usr/include/project.h
632 file path=usr/include/protocols/dumprestore.h
633 file path=usr/include/protocols/routed.h
634 file path=usr/include/protocols/rwhod.h
635 file path=usr/include/protocols/timed.h
636 file path=usr/include/pthread.h
637 file path=usr/include/pw.h
638 file path=usr/include/pwd.h
639 file path=usr/include/rcm_module.h
640 file path=usr/include/rctl.h
641 file path=usr/include/re_comp.h
642 file path=usr/include/regex.h
643 file path=usr/include/regexp.h
644 file path=usr/include/regexpr.h
645 file path=usr/include/resolv.h
646 file path=usr/include/rje.h
647 file path=usr/include/rp_plugin.h
648 file path=usr/include/rpc/auth.h
649 file path=usr/include/rpc/auth_des.h
650 file path=usr/include/rpc/auth_sys.h
651 file path=usr/include/rpc/auth_unix.h
652 file path=usr/include/rpc/bootparam.h
653 file path=usr/include/rpc/clnt.h
```

```
file path=usr/include/rpc/clnt_soc.h
file path=usr/include/rpc/clnt_stat.h
file path=usr/include/rpc/des_crypt.h
$(sparc_ONLY)file path=usr/include/rpc/ib.h
file path=usr/include/rpc/key_prot.h
file path=usr/include/rpc/nettype.h
file path=usr/include/rpc/pmap_clnt.h
file path=usr/include/rpc/pmap_prot.h
file path=usr/include/rpc/pmap_prot.x
file path=usr/include/rpc/pmap_rmt.h
file path=usr/include/rpc/raw.h
file path=usr/include/rpc/rpc.h
file path=usr/include/rpc/rpc_com.h
file path=usr/include/rpc/rpc_msg.h
file path=usr/include/rpc/rpc_rdma.h
file path=usr/include/rpc/rpc_sztypes.h
file path=usr/include/rpc/rpcb_clnt.h
file path=usr/include/rpc/rpcb_prot.h
file path=usr/include/rpc/rpcb_prot.x
file path=usr/include/rpc/rpcent.h
file path=usr/include/rpc/rpcsec_gss.h
file path=usr/include/rpc/rpcsys.h
file path=usr/include/rpc/svc.h
file path=usr/include/rpc/svc_auth.h
file path=usr/include/rpc/svc_mt.h
file path=usr/include/rpc/svc_soc.h
file path=usr/include/rpc/types.h
file path=usr/include/rpc/xdr.h
file path=usr/include/rpcsvc/autofs_prot.h
file path=usr/include/rpcsvc/autofs_prot.x
file path=usr/include/rpcsvc/bootparam.h
file path=usr/include/rpcsvc/bootparam_prot.h
file path=usr/include/rpcsvc/bootparam_prot.x
file path=usr/include/rpcsvc/dbm.h
file path=usr/include/rpcsvc/key_prot.x
file path=usr/include/rpcsvc/mount.h
file path=usr/include/rpcsvc/mount.x
file path=usr/include/rpcsvc/nfs4_prot.h
file path=usr/include/rpcsvc/nfs4_prot.x
file path=usr/include/rpcsvc/nfs_acl.h
file path=usr/include/rpcsvc/nfs_acl.x
file path=usr/include/rpcsvc/nfs_prot.h
file path=usr/include/rpcsvc/nfs_prot.x
file path=usr/include/rpcsvc/nis.h
file path=usr/include/rpcsvc/nis.x
file path=usr/include/rpcsvc/nis_db.h
file path=usr/include/rpcsvc/nis_object.x
file path=usr/include/rpcsvc/nislib.h
file path=usr/include/rpcsvc/nlm_prot.h
file path=usr/include/rpcsvc/nlm_prot.x
file path=usr/include/rpcsvc/nsm_addr.h
file path=usr/include/rpcsvc/nsm_addr.x
file path=usr/include/rpcsvc/rex.h
file path=usr/include/rpcsvc/rex.x
file path=usr/include/rpcsvc/rpc_sztypes.h
file path=usr/include/rpcsvc/rpc_sztypes.x
file path=usr/include/rpcsvc/rquota.h
file path=usr/include/rpcsvc/rquota.x
file path=usr/include/rpcsvc/rstat.h
file path=usr/include/rpcsvc/rstat.x
file path=usr/include/rpcsvc/rusers.h
file path=usr/include/rpcsvc/rusers.x
file path=usr/include/rpcsvc/rwall.h
file path=usr/include/rpcsvc/rwall.x
file path=usr/include/rpcsvc/sm_inter.h
file path=usr/include/rpcsvc/sm_inter.x
```



```
file path=usr/include/rpcsvc/spray.h
file path=usr/include/rpcsvc/spray.x
file path=usr/include/rpcsvc/ufs_prot.h
file path=usr/include/rpcsvc/ufs_prot.x
file path=usr/include/rpcsvc/yp.x
file path=usr/include/rpcsvc/yp_prot.h
file path=usr/include/rpcsvc/ypclnt.h
file path=usr/include/rpcsvc/yppasswd.h
file path=usr/include/rpcsvc/ypupd.h
file path=usr/include/rsmapi.h
file path=usr/include/rtld_db.h
file path=usr/include/sac.h
file path=usr/include/sasl/prop.h
file path=usr/include/sasl/sasl.h
file path=usr/include/sasl/saslplug.h
file path=usr/include/sasl/saslutil.h
file path=usr/include/sched.h
file path=usr/include/schedctl.h
file path=usr/include/scsi/libscsi.h
file path=usr/include/scsi/libses.h
file path=usr/include/scsi/libses_plugin.h
file path=usr/include/scsi/libsmp.h
file path=usr/include/scsi/libsmp_plugin.h
file path=usr/include/scsi/plugins/ses/framework/libses.h
file path=usr/include/scsi/plugins/ses/framework/ses2.h
file path=usr/include/scsi/plugins/ses/framework/ses2_impl.h
file path=usr/include/scsi/plugins/ses/vendor/sun.h
file path=usr/include/sdp.h
file path=usr/include/search.h
file path=usr/include/secdb.h
file path=usr/include/security/auditd.h
file path=usr/include/security/cryptoki.h
file path=usr/include/security/pam_appl.h
file path=usr/include/security/pam_modules.h
file path=usr/include/security/pkcs11.h
file path=usr/include/security/pkcs11f.h
file path=usr/include/security/pkcs11t.h
file path=usr/include/semaphore.h
file path=usr/include/setjmp.h
file path=usr/include/sgtty.h
file path=usr/include/sha1.h
file path=usr/include/sha2.h
file path=usr/include/shadow.h
file path=usr/include/sharefs/share.h
file path=usr/include/sharefs/sharefs.h
file path=usr/include/sharefs/sharetab.h
file path=usr/include/siginfo.h
file path=usr/include/signal.h
file path=usr/include/sip.h
file path=usr/include/smbios.h
file path=usr/include/spawn.h
$(i386_ONLY)file path=usr/include/stack_unwind.h
file path=usr/include/stdarg.h
file path=usr/include/stdbool.h
file path=usr/include/stddef.h
file path=usr/include/stdint.h
file path=usr/include/stdio.h
file path=usr/include/stdio_ext.h
file path=usr/include/stdio_impl.h
file path=usr/include/stdio_tag.h
file path=usr/include/stdlib.h
file path=usr/include/storclass.h
file path=usr/include/string.h
file path=usr/include/strings.h
file path=usr/include/stropts.h
file path=usr/include/syms.h
```

```
786 file path=usr/include/synch.h
787 file path=usr/include/sys/acct.h
788 file path=usr/include/sys/acctctl.h
789 file path=usr/include/sys/acl.h
790 file path=usr/include/sys/acl_impl.h
791 file path=usr/include/sys/acpi_drv.h
792 file path=usr/include/sys/aio.h
793 file path=usr/include/sys/aio_impl.h
794 file path=usr/include/sys/aio_req.h
795 file path=usr/include/sys/aiocb.h
796 file path=usr/include/sys/archsystm.h
797 file path=usr/include/sys/ascii.h
798 file path=usr/include/sys/asm_linkage.h
799 file path=usr/include/sys/asynch.h
800 file path=usr/include/sys/atomic.h
801 file path=usr/include/sys/attr.h
802 file path=usr/include/sys/autoconf.h
803 file path=usr/include/sys/auxv.h
804 file path=usr/include/sys/auxv_386.h
805 file path=usr/include/sys/auxv_SPARC.h
806 file path=usr/include/sys/av/iec61883.h
807 file path=usr/include/sys/avintr.h
808 file path=usr/include/sys/avl.h
809 file path=usr/include/sys/avl_impl.h
810 file path=usr/include/sys/bitmap.h
811 file path=usr/include/sys/bitset.h
812 file path=usr/include/sys/bl.h
813 file path=usr/include/sys/blkdev.h
814 file path=usr/include/sys/bofi.h
815 file path=usr/include/sys/bofi_impl.h
816 file path=usr/include/sys/bootconf.h
817 $(i386_ONLY)file path=usr/include/sys/bootregs.h
818 file path=usr/include/sys/bootstat.h
819 $(i386_ONLY)file path=usr/include/sys/bootsvcs.h
820 file path=usr/include/sys/bpp_io.h
821 file path=usr/include/sys/brand.h
822 file path=usr/include/sys/buf.h
823 file path=usr/include/sys/bufmod.h
824 file path=usr/include/sys/bustypes.h
825 file path=usr/include/sys/byteorder.h
826 file path=usr/include/sys/callb.h
827 file path=usr/include/sys/callo.h
828 file path=usr/include/sys/cap_util.h
829 file path=usr/include/sys/ccompile.h
830 file path=usr/include/sys/cdio.h
831 file path=usr/include/sys/cis.h
832 file path=usr/include/sys/cis_handlers.h
833 file path=usr/include/sys/cis_protos.h
834 file path=usr/include/sys/cladm.h
835 file path=usr/include/sys/class.h
836 file path=usr/include/sys/clconf.h
837 file path=usr/include/sys/cmlb.h
838 file path=usr/include/sys/cmn_err.h
839 $(sparc_ONLY)file path=usr/include/sys/cmpregs.h
840 file path=usr/include/sys/compress.h
841 file path=usr/include/sys/condvar.h
842 file path=usr/include/sys/condvar_impl.h
843 file path=usr/include/sys/conf.h
844 file path=usr/include/sys/consdev.h
845 file path=usr/include/sys/console.h
846 file path=usr/include/sys/consplat.h
847 file path=usr/include/sys/contract.h
848 file path=usr/include/sys/contract/device.h
849 file path=usr/include/sys/contract/device_impl.h
850 file path=usr/include/sys/contract/process.h
851 file path=usr/include/sys/contract/process_impl.h
```



```
852 file path=usr/include/sys/contract_impl.h
853 $(i386_ONLY)file path=usr/include/sys/controlregs.h
854 file path=usr/include/sys/copyops.h
855 file path=usr/include/sys/core.h
856 file path=usr/include/sys/corectl.h
857 file path=usr/include/sys/cpc_impl.h
858 file path=usr/include/sys/cpc_pcbe.h
859 file path=usr/include/sys/cpr.h
860 file path=usr/include/sys/cpu.h
861 file path=usr/include/sys/cpucaps.h
862 file path=usr/include/sys/cpucaps_impl.h
863 file path=usr/include/sys/cpupart.h
864 file path=usr/include/sys/cpuvar.h
865 file path=usr/include/sys/crc32.h
866 file path=usr/include/sys/cred.h
867 file path=usr/include/sys/cred_impl.h
868 file path=usr/include/sys/crtctl.h
869 file path=usr/include/sys/crypto/api.h
870 file path=usr/include/sys/crypto/common.h
871 file path=usr/include/sys/crypto/ioctl.h
872 file path=usr/include/sys/crypto/ioctladmin.h
873 file path=usr/include/sys/crypto/spi.h
874 file path=usr/include/sys/cs.h
875 file path=usr/include/sys/cs_priv.h
876 file path=usr/include/sys/cs_strings.h
877 file path=usr/include/sys/cs_stubs.h
878 file path=usr/include/sys/cs_types.h
879 file path=usr/include/sys/csiioctl.h
880 file path=usr/include/sys/ctf.h
881 file path=usr/include/sys/ctf_api.h
882 file path=usr/include/sys/ctfs.h
883 file path=usr/include/sys/ctfs_impl.h
884 file path=usr/include/sys/ctype.h
885 file path=usr/include/sys/cyclic.h
886 file path=usr/include/sys/cyclic_impl.h
887 file path=usr/include/sys/dacf.h
888 file path=usr/include/sys/dacf_impl.h
889 file path=usr/include/sys/damap.h
890 file path=usr/include/sys/damap_impl.h
891 file path=usr/include/sys/dc_ki.h
892 file path=usr/include/sys/ddi.h
893 file path=usr/include/sys/ddi_hp.h
894 file path=usr/include/sys/ddi_hp_impl.h
895 file path=usr/include/sys/ddi_impldefs.h
896 file path=usr/include/sys/ddi_implfuncs.h
897 file path=usr/include/sys/ddi_intr.h
898 file path=usr/include/sys/ddi_intr_impl.h
899 file path=usr/include/sys/ddi_isa.h
900 file path=usr/include/sys/ddi_obsolete.h
901 file path=usr/include/sys/ddi_periodic.h
902 file path=usr/include/sys/ddidevmap.h
903 file path=usr/include/sys/ddidmareq.h
904 file path=usr/include/sys/ddifm.h
905 file path=usr/include/sys/ddifm_impl.h
906 file path=usr/include/sys/ddimapreq.h
907 file path=usr/include/sys/ddipropdefs.h
908 file path=usr/include/sys/dditypes.h
909 file path=usr/include/sys/debug.h
910 $(i386_ONLY)file path=usr/include/sys/debugreg.h
911 file path=usr/include/sys/des.h
912 file path=usr/include/sys/devcache.h
913 file path=usr/include/sys/devcache_impl.h
914 file path=usr/include/sys/devctl.h
915 file path=usr/include/sys/devfm.h
916 file path=usr/include/sys/devid_cache.h
917 file path=usr/include/sys/devinfo_impl.h
```

```
 918 file path=usr/include/sys/devops.h
 919 file path=usr/include/sys/devpolicy.h
 920 file path=usr/include/sys/devpoll.h
 921 file path=usr/include/sys/dirent.h
 922 file path=usr/include/sys/disp.h
 923 file path=usr/include/sys/dkbad.h
 924 file path=usr/include/sys/dkio.h
 925 file path=usr/include/sys/dklabel.h
 926 $(sparc_ONLY)file path=usr/include/sys/dkmpio.h
 927 $(i386_ONLY)file path=usr/include/sys/dktp/altsctr.h
 928 $(i386_ONLY)file path=usr/include/sys/dktp/cmpkt.h
 929 file path=usr/include/sys/dktp/dadkio.h
 930 file path=usr/include/sys/dktp/fdisk.h
 931 file path=usr/include/sys/dl.h
 932 file path=usr/include/sys/dld.h
 933 file path=usr/include/sys/dlpi.h
 934 file path=usr/include/sys/dls_mgmt.h
 935 $(i386_ONLY)file path=usr/include/sys/dma_engine.h
 936 file path=usr/include/sys/dma_i8237A.h
 937 file path=usr/include/sys/dnlc.h
 938 file path=usr/include/sys/door.h
 939 file path=usr/include/sys/door_data.h
 940 file path=usr/include/sys/door_impl.h
 941 file path=usr/include/sys/dumphdr.h
 942 file path=usr/include/sys/ecppio.h
 943 file path=usr/include/sys/ecppreg.h
 944 file path=usr/include/sys/ecppsys.h
 945 file path=usr/include/sys/ecppvar.h
 946 file path=usr/include/sys/efi_partition.h
 947 file path=usr/include/sys/elf.h
 948 file path=usr/include/sys/elf_386.h
 949 file path=usr/include/sys/elf_SPARC.h
 950 file path=usr/include/sys/elf_amd64.h
 951 file path=usr/include/sys/elf_notes.h
 952 file path=usr/include/sys/elftypes.h
 953 file path=usr/include/sys/epm.h
 954 file path=usr/include/sys/errno.h
 955 file path=usr/include/sys/errorq.h
 956 file path=usr/include/sys/errorq_impl.h
 957 file path=usr/include/sys/esunddi.h
 958 file path=usr/include/sys/ethernet.h
 959 file path=usr/include/sys/euc.h
 960 file path=usr/include/sys/eucioctl.h
 961 file path=usr/include/sys/exacct.h
 962 file path=usr/include/sys/exacct_catalog.h
 963 file path=usr/include/sys/exacct_impl.h
 964 file path=usr/include/sys/exec.h
 965 file path=usr/include/sys/exechdr.h
 966 file path=usr/include/sys/fault.h
 967 file path=usr/include/sys/fbio.h
 968 file path=usr/include/sys/fbuf.h
 969 file path=usr/include/sys/fc4/fc.h
 970 file path=usr/include/sys/fc4/fc_transport.h
 971 file path=usr/include/sys/fc4/fcal.h
 972 file path=usr/include/sys/fc4/fcal_linkapp.h
 973 file path=usr/include/sys/fc4/fcal_transport.h
 974 file path=usr/include/sys/fc4/fcio.h
 975 file path=usr/include/sys/fc4/fcp.h
 976 file path=usr/include/sys/fc4/linkapp.h
 977 file path=usr/include/sys/fcntl.h
 978 file path=usr/include/sys/fdbuffer.h
 979 file path=usr/include/sys/fdio.h
 980 $(sparc_ONLY)file path=usr/include/sys/fdreg.h
 981 $(sparc_ONLY)file path=usr/include/sys/fdvar.h
 982 file path=usr/include/sys/feature_tests.h
 983 file path=usr/include/sys/fem.h
```



```
 984 file path=usr/include/sys/file.h
 985 file path=usr/include/sys/filio.h
 986 file path=usr/include/sys/flock.h
 987 file path=usr/include/sys/flock_impl.h
 988 $(sparc_ONLY)file path=usr/include/sys/fm/cpu/SPARC64-VI.h
 989 $(sparc_ONLY)file path=usr/include/sys/fm/cpu/UltraSPARC-II.h
 990 $(sparc_ONLY)file path=usr/include/sys/fm/cpu/UltraSPARC-III.h
 991 $(sparc_ONLY)file path=usr/include/sys/fm/cpu/UltraSPARC-T1.h
 992 file path=usr/include/sys/fm/fs/zfs.h
 993 file path=usr/include/sys/fm/io/ddi.h
 994 file path=usr/include/sys/fm/io/disk.h
 995 file path=usr/include/sys/fm/io/opl_mc_fm.h
 996 file path=usr/include/sys/fm/io/pci.h
 997 file path=usr/include/sys/fm/io/scsi.h
 998 file path=usr/include/sys/fm/io/sun4upci.h
 999 file path=usr/include/sys/fm/protocol.h
1000 file path=usr/include/sys/fm/util.h
1001 file path=usr/include/sys/fork.h
1002 $(i386_ONLY)file path=usr/include/sys/fp.h
1003 $(sparc_ONLY)file path=usr/include/sys/fpu/fpu_simulator.h
1004 $(sparc_ONLY)file path=usr/include/sys/fpu/fpusystm.h
1005 $(sparc_ONLY)file path=usr/include/sys/fpu/globals.h
1006 $(sparc_ONLY)file path=usr/include/sys/fpu/ieee.h
1007 file path=usr/include/sys/frame.h
1008 file path=usr/include/sys/fs/autofs.h
1009 file path=usr/include/sys/fs/cachefs_dir.h
1010 file path=usr/include/sys/fs/cachefs_dlog.h
1011 file path=usr/include/sys/fs/cachefs_filegrp.h
1012 file path=usr/include/sys/fs/cachefs_fs.h
1013 file path=usr/include/sys/fs/cachefs_fscache.h
1014 file path=usr/include/sys/fs/cachefs_ioctl.h
1015 file path=usr/include/sys/fs/cachefs_log.h
1016 file path=usr/include/sys/fs/decomp.h
1017 file path=usr/include/sys/fs/dv_node.h
1018 file path=usr/include/sys/fs/fifonode.h
1019 file path=usr/include/sys/fs/hsfs_isospec.h
1020 file path=usr/include/sys/fs/hsfs_node.h
1021 file path=usr/include/sys/fs/hsfs_rrip.h
1022 file path=usr/include/sys/fs/hsfs_spec.h
1023 file path=usr/include/sys/fs/hsfs_susp.h
1024 file path=usr/include/sys/fs/lofs_info.h
1025 file path=usr/include/sys/fs/lofs_node.h
1026 file path=usr/include/sys/fs/mntdata.h
1027 file path=usr/include/sys/fs/namenode.h
1028 file path=usr/include/sys/fs/pc_dir.h
1029 file path=usr/include/sys/fs/pc_fs.h
1030 file path=usr/include/sys/fs/pc_label.h
1031 file path=usr/include/sys/fs/pc_node.h
1032 file path=usr/include/sys/fs/pxfs_ki.h
1033 file path=usr/include/sys/fs/sdev_impl.h
1034 file path=usr/include/sys/fs/snode.h
1035 file path=usr/include/sys/fs/swapnode.h
1036 file path=usr/include/sys/fs/tmp.h
1037 file path=usr/include/sys/fs/tmpnode.h
1038 file path=usr/include/sys/fs/udf_inode.h
1039 file path=usr/include/sys/fs/udf_volume.h
1040 file path=usr/include/sys/fs/ufs_acl.h
1041 file path=usr/include/sys/fs/ufs_bio.h
1042 file path=usr/include/sys/fs/ufs_filio.h
1043 file path=usr/include/sys/fs/ufs_fs.h
1044 file path=usr/include/sys/fs/ufs_fsdir.h
1045 file path=usr/include/sys/fs/ufs_inode.h
1046 file path=usr/include/sys/fs/ufs_lockfs.h
1047 file path=usr/include/sys/fs/ufs_log.h
1048 file path=usr/include/sys/fs/ufs_mount.h
1049 file path=usr/include/sys/fs/ufs_panic.h
```

```
1050 file path=usr/include/sys/fs/ufs_prot.h
1051 file path=usr/include/sys/fs/ufs_quota.h
1052 file path=usr/include/sys/fs/ufs_snap.h
1053 file path=usr/include/sys/fs/ufs_trans.h
1054 file path=usr/include/sys/fs/zfs.h
1055 file path=usr/include/sys/fs_reparse.h
1056 file path=usr/include/sys/fs_subr.h
1057 file path=usr/include/sys/fsid.h
1058 $(sparc_ONLY)file path=usr/include/sys/fsr.h
1059 file path=usr/include/sys/fss.h
1060 file path=usr/include/sys/fssnap.h
1061 file path=usr/include/sys/fssnap_if.h
1062 file path=usr/include/sys/fsspriocntl.h
1063 file path=usr/include/sys/fstyp.h
1064 file path=usr/include/sys/ftrace.h
1065 file path=usr/include/sys/fx.h
1066 file path=usr/include/sys/fxpriocntl.h
1067 file path=usr/include/sys/gfs.h
1068 file path=usr/include/sys/gld.h
1069 file path=usr/include/sys/gldpriv.h
1070 file path=usr/include/sys/group.h
1071 file path=usr/include/sys/hdio.h
1072 file path=usr/include/sys/hook.h
1073 file path=usr/include/sys/hook_event.h
1074 file path=usr/include/sys/hook_impl.h
1075 file path=usr/include/sys/hotplug/hpcsvc.h
1076 file path=usr/include/sys/hotplug/hpctrl.h
1077 file path=usr/include/sys/hotplug/pci/pcicfg.h
1078 file path=usr/include/sys/hotplug/pci/pcihp.h
1079 file path=usr/include/sys/hwconf.h
1080 $(i386_ONLY)file path=usr/include/sys/hypervisor.h
1081 $(i386_ONLY)file path=usr/include/sys/i8272A.h
1082 file path=usr/include/sys/ia.h
1083 file path=usr/include/sys/iapriocntl.h
1084 file path=usr/include/sys/ib/adapters/hermon/hermon_ioctl.h
1085 file path=usr/include/sys/ib/adapters/mlnx_umap.h
1086 file path=usr/include/sys/ib/adapters/tavor/tavor_ioctl.h
1087 file path=usr/include/sys/ib/clients/ibd/ibd.h
1088 file path=usr/include/sys/ib/clients/of/ofa_solaris.h
1089 file path=usr/include/sys/ib/clients/of/ofed_kernel.h
1090 file path=usr/include/sys/ib/clients/of/rdma/ib_addr.h
1091 file path=usr/include/sys/ib/clients/of/rdma/ib_user_mad.h
1092 file path=usr/include/sys/ib/clients/of/rdma/ib_user_sa.h
1093 file path=usr/include/sys/ib/clients/of/rdma/ib_user_verbs.h
1094 file path=usr/include/sys/ib/clients/of/rdma/ib_verbs.h
1095 file path=usr/include/sys/ib/clients/of/rdma/rdma_cm.h
1096 file path=usr/include/sys/ib/clients/of/rdma/rdma_user_cm.h
1097 file path=usr/include/sys/ib/clients/of/sol_ofs/sol_cma.h
1098 file path=usr/include/sys/ib/clients/of/sol_ofs/sol_ib_cma.h
1099 file path=usr/include/sys/ib/clients/of/sol_ofs/sol_kverb_impl.h
1100 file path=usr/include/sys/ib/clients/of/sol_ofs/sol_ofs_common.h
1101 file path=usr/include/sys/ib/clients/of/sol_ucma/sol_rdma_user_cm.h
1102 file path=usr/include/sys/ib/clients/of/sol_ucma/sol_ucma.h
1103 file path=usr/include/sys/ib/clients/of/sol_umad/sol_umad.h
1104 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs.h
1105 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs2ucma.h
1106 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs_comp.h
1107 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs_event.h
1108 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs_hca.h
1109 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs_qp.h
1110 file path=usr/include/sys/ib/ib_pkt_hdrs.h
1111 file path=usr/include/sys/ib/ib_types.h
1112 file path=usr/include/sys/ib/ibnex/ibnex_devctl.h
1113 file path=usr/include/sys/ib/ibtl/ibci.h
1114 file path=usr/include/sys/ib/ibtl/ibti.h
1115 file path=usr/include/sys/ib/ibtl/ibti_cm.h
```



```
1116 file path=usr/include/sys/ib/ibtl/ibti_common.h
1117 file path=usr/include/sys/ib/ibtl/ibtl_ci_types.h
1118 file path=usr/include/sys/ib/ibtl/ibtl_status.h
1119 file path=usr/include/sys/ib/ibtl/ibtl_types.h
1120 file path=usr/include/sys/ib/ibtl/ibvti.h
1121 file path=usr/include/sys/ib/ibtl/impl/ibtl_util.h
1122 file path=usr/include/sys/ib/mgt/ib_dm_attr.h
1123 file path=usr/include/sys/ib/mgt/ib_mad.h
1124 file path=usr/include/sys/ib/mgt/ibmf/ibmf.h
1125 file path=usr/include/sys/ib/mgt/ibmf/ibmf_msg.h
1126 file path=usr/include/sys/ib/mgt/ibmf/ibmf_saa.h
1127 file path=usr/include/sys/ib/mgt/ibmf/ibmf_utils.h
1128 file path=usr/include/sys/ib/mgt/sa_recs.h
1129 file path=usr/include/sys/ib/mgt/sm_attr.h
1130 file path=usr/include/sys/ibpart.h
1131 file path=usr/include/sys/id32.h
1132 file path=usr/include/sys/id_space.h
1133 file path=usr/include/sys/idmap.h
1134 file path=usr/include/sys/inline.h
1135 file path=usr/include/sys/instance.h
1136 file path=usr/include/sys/int_const.h
1137 file path=usr/include/sys/int_fmtio.h
1138 file path=usr/include/sys/int_limits.h
1139 file path=usr/include/sys/int_types.h
1140 file path=usr/include/sys/inttypes.h
1141 file path=usr/include/sys/ioccom.h
1142 file path=usr/include/sys/ioctl.h
1143 $(i386_ONLY)file path=usr/include/sys/iommulib.h
1144 file path=usr/include/sys/ipc.h
1145 file path=usr/include/sys/ipc_impl.h
1146 file path=usr/include/sys/ipc_rctl.h
1147 file path=usr/include/sys/isa_defs.h
1148 file path=usr/include/sys/iso/signal_iso.h
1149 file path=usr/include/sys/jioctl.h
1150 file path=usr/include/sys/kbd.h
1151 file path=usr/include/sys/kbdreg.h
1152 file path=usr/include/sys/kbio.h
1153 file path=usr/include/sys/kcpc.h
1154 file path=usr/include/sys/kd.h
1155 file path=usr/include/sys/kdi.h
1156 file path=usr/include/sys/kdi_impl.h
1157 file path=usr/include/sys/kdi_machimpl.h
1158 $(i386_ONLY)file path=usr/include/sys/kdi_regs.h
1159 file path=usr/include/sys/kiconv.h
1160 file path=usr/include/sys/kidmap.h
1161 file path=usr/include/sys/klpd.h
1162 file path=usr/include/sys/klwp.h
1163 file path=usr/include/sys/kmem.h
1164 file path=usr/include/sys/kmem_impl.h
1165 file path=usr/include/sys/kobj.h
1166 file path=usr/include/sys/kobj_impl.h
1167 file path=usr/include/sys/ksocket.h
1168 file path=usr/include/sys/kstat.h
1169 file path=usr/include/sys/kstr.h
1170 file path=usr/include/sys/ksyms.h
1171 file path=usr/include/sys/ksynch.h
1172 file path=usr/include/sys/lc_core.h
1173 file path=usr/include/sys/ldterm.h
1174 file path=usr/include/sys/lgrp.h
1175 file path=usr/include/sys/lgrp_user.h
1176 file path=usr/include/sys/link.h
1177 file path=usr/include/sys/list.h
1178 file path=usr/include/sys/list_impl.h
1179 file path=usr/include/sys/llc1.h
1180 file path=usr/include/sys/loadavg.h
1181 file path=usr/include/sys/localedef.h
```

```
1182 file path=usr/include/sys/lock.h
1183 file path=usr/include/sys/lockfs.h
1184 file path=usr/include/sys/lofi.h
1185 file path=usr/include/sys/log.h
1186 file path=usr/include/sys/logindmux.h
1187 file path=usr/include/sys/lvm/md_basic.h
1188 file path=usr/include/sys/lvm/md_convert.h
1189 file path=usr/include/sys/lvm/md_crc.h
1190 file path=usr/include/sys/lvm/md_hotspares.h
1191 file path=usr/include/sys/lvm/md_mddb.h
1192 file path=usr/include/sys/lvm/md_mdiox.h
1193 file path=usr/include/sys/lvm/md_mhdx.h
1194 file path=usr/include/sys/lvm/md_mirror.h
1195 file path=usr/include/sys/lvm/md_mirror_shared.h
1196 file path=usr/include/sys/lvm/md_names.h
1197 file path=usr/include/sys/lvm/md_notify.h
1198 file path=usr/include/sys/lvm/md_raid.h
1199 file path=usr/include/sys/lvm/md_rename.h
1200 file path=usr/include/sys/lvm/md_sp.h
1201 file path=usr/include/sys/lvm/md_stripe.h
1202 file path=usr/include/sys/lvm/md_trans.h
1203 file path=usr/include/sys/lvm/mdio.h
1204 file path=usr/include/sys/lvm/mdmed.h
1205 file path=usr/include/sys/lvm/mdmn_commd.h
1206 file path=usr/include/sys/lvm/mdvar.h
1207 file path=usr/include/sys/lwp.h
1208 file path=usr/include/sys/lwp_timer_impl.h
1209 file path=usr/include/sys/lwp_upimutex_impl.h
1210 file path=usr/include/sys/mac.h
1211 file path=usr/include/sys/mac_ether.h
1212 file path=usr/include/sys/mac_flow.h
1213 file path=usr/include/sys/mac_provider.h
1214 file path=usr/include/sys/machelf.h
1215 file path=usr/include/sys/machlock.h
1216 file path=usr/include/sys/machsig.h
1217 file path=usr/include/sys/machtypes.h
1218 file path=usr/include/sys/map.h
1219 $(i386_ONLY)file path=usr/include/sys/mc.h
1220 $(i386_ONLY)file path=usr/include/sys/mc_amd.h
1221 $(i386_ONLY)file path=usr/include/sys/mc_intel.h
1222 $(i386_ONLY)file path=usr/include/sys/mca_amd.h
1223 $(i386_ONLY)file path=usr/include/sys/mca_x86.h
1224 file path=usr/include/sys/md4.h
1225 file path=usr/include/sys/md5.h
1226 file path=usr/include/sys/md5_consts.h
1227 file path=usr/include/sys/mdi_impldefs.h
1228 file path=usr/include/sys/mem.h
1229 file path=usr/include/sys/mem_config.h
1230 file path=usr/include/sys/memlist.h
1231 file path=usr/include/sys/mhd.h
1232 file path=usr/include/sys/mii.h
1233 file path=usr/include/sys/miiregs.h
1234 file path=usr/include/sys/mkdev.h
1235 file path=usr/include/sys/mman.h
1236 file path=usr/include/sys/mmapobj.h
1237 file path=usr/include/sys/mntent.h
1238 file path=usr/include/sys/mntio.h
1239 file path=usr/include/sys/mnttab.h
1240 file path=usr/include/sys/modctl.h
1241 file path=usr/include/sys/mode.h
1242 file path=usr/include/sys/model.h
1243 file path=usr/include/sys/modhash.h
1244 file path=usr/include/sys/modhash_impl.h
1245 file path=usr/include/sys/mount.h
1246 file path=usr/include/sys/mouse.h
1247 file path=usr/include/sys/msacct.h
```



```
1248 file path=usr/include/sys/msg.h
1249 file path=usr/include/sys/msg_impl.h
1250 file path=usr/include/sys/msio.h
1251 file path=usr/include/sys/msreg.h
1252 file path=usr/include/sys/mtio.h
1253 file path=usr/include/sys/multidata.h
1254 file path=usr/include/sys/mutex.h
1255 $(i386_ONLY)file path=usr/include/sys/mutex_impl.h
1256 file path=usr/include/sys/nbmlock.h
1257 file path=usr/include/sys/ndi_impldefs.h
1258 file path=usr/include/sys/ndifm.h
1259 file path=usr/include/sys/netconfig.h
1260 file path=usr/include/sys/neti.h
1261 file path=usr/include/sys/netstack.h
1262 file path=usr/include/sys/nexusdefs.h
1263 file path=usr/include/sys/note.h
1264 file path=usr/include/sys/null.h
1265 file path=usr/include/sys/nvpair.h
1266 file path=usr/include/sys/nvpair_impl.h
1267 file path=usr/include/sys/objfs.h
1268 file path=usr/include/sys/objfs_impl.h
1269 file path=usr/include/sys/obpdefs.h
1270 file path=usr/include/sys/old_procfs.h
1271 file path=usr/include/sys/open.h
1272 file path=usr/include/sys/openpromio.h
1273 file path=usr/include/sys/panic.h
1274 file path=usr/include/sys/param.h
1275 file path=usr/include/sys/pathconf.h
1276 file path=usr/include/sys/pathname.h
1277 file path=usr/include/sys/pattr.h
1278 file path=usr/include/sys/pbio.h
1279 file path=usr/include/sys/pcb.h
1280 file path=usr/include/sys/pccard.h
1281 file path=usr/include/sys/pci.h
1282 $(i386_ONLY)file path=usr/include/sys/pcic_reg.h
1283 $(i386_ONLY)file path=usr/include/sys/pcic_var.h
1284 file path=usr/include/sys/pcie.h
1285 file path=usr/include/sys/pcmcia.h
1286 file path=usr/include/sys/pctypes.h
1287 file path=usr/include/sys/pfmod.h
1288 file path=usr/include/sys/pg.h
1289 file path=usr/include/sys/pghw.h
1290 file path=usr/include/sys/physmem.h
1291 $(i386_ONLY)file path=usr/include/sys/pic.h
1292 $(i386_ONLY)file path=usr/include/sys/pit.h
1293 file path=usr/include/sys/pkp_hash.h
1294 file path=usr/include/sys/pm.h
1295 $(i386_ONLY)file path=usr/include/sys/pmem.h
1296 file path=usr/include/sys/policy.h
1297 file path=usr/include/sys/poll.h
1298 file path=usr/include/sys/poll_impl.h
1299 file path=usr/include/sys/pool.h
1300 file path=usr/include/sys/pool_impl.h
1301 file path=usr/include/sys/pool_pset.h
1302 file path=usr/include/sys/port.h
1303 file path=usr/include/sys/port_impl.h
1304 file path=usr/include/sys/port_kernel.h
1305 file path=usr/include/sys/ppmio.h
1306 file path=usr/include/sys/priocntl.h
1307 file path=usr/include/sys/priv.h
1308 file path=usr/include/sys/priv_const.h
1309 file path=usr/include/sys/priv_impl.h
1310 file path=usr/include/sys/priv_names.h
1311 $(i386_ONLY)file path=usr/include/sys/privmregs.h
1312 $(i386_ONLY)file path=usr/include/sys/privregs.h
1313 file path=usr/include/sys/prnio.h
```

```
1314 file path=usr/include/sys/proc.h
1315 file path=usr/include/sys/proc/prdata.h
1316 file path=usr/include/sys/processor.h
1317 file path=usr/include/sys/procfs.h
1318 file path=usr/include/sys/procfs_isa.h
1319 file path=usr/include/sys/procset.h
1320 file path=usr/include/sys/project.h
1321 $(i386_ONLY)file path=usr/include/sys/prom_emul.h
1322 $(i386_ONLY)file path=usr/include/sys/prom_isa.h
1323 $(i386_ONLY)file path=usr/include/sys/prom_plat.h
1324 file path=usr/include/sys/promif.h
1325 file path=usr/include/sys/promimpl.h
1326 file path=usr/include/sys/protosw.h
1327 file path=usr/include/sys/prsystm.h
1328 file path=usr/include/sys/pset.h
1329 file path=usr/include/sys/psw.h
1330 $(i386_ONLY)file path=usr/include/sys/pte.h
1331 file path=usr/include/sys/ptem.h
1332 file path=usr/include/sys/ptms.h
1333 file path=usr/include/sys/ptyvar.h
1334 file path=usr/include/sys/queue.h
1335 file path=usr/include/sys/raidioctl.h
1336 file path=usr/include/sys/ramdisk.h
1337 file path=usr/include/sys/random.h
1338 file path=usr/include/sys/rctl.h
1339 file path=usr/include/sys/rctl_impl.h
1340 file path=usr/include/sys/rds.h
1341 file path=usr/include/sys/reboot.h
1342 file path=usr/include/sys/refstr.h
1343 file path=usr/include/sys/refstr_impl.h
1344 file path=usr/include/sys/reg.h
1345 file path=usr/include/sys/regset.h
1346 file path=usr/include/sys/resource.h
1347 file path=usr/include/sys/rlioctl.h
1348 file path=usr/include/sys/rsm/rsm.h
1349 file path=usr/include/sys/rsm/rsm_common.h
1350 file path=usr/include/sys/rsm/rsmapi_common.h
1351 file path=usr/include/sys/rsm/rsmka_path_int.h
1352 file path=usr/include/sys/rsm/rsmndi.h
1353 file path=usr/include/sys/rsm/rsmpi.h
1354 file path=usr/include/sys/rsm/rsmpi_driver.h
1355 file path=usr/include/sys/rt.h
1356 $(i386_ONLY)file path=usr/include/sys/rtc.h
1357 file path=usr/include/sys/rtpriocntl.h
1358 file path=usr/include/sys/rwlock.h
1359 file path=usr/include/sys/rwlock_impl.h
1360 file path=usr/include/sys/rwstlock.h
1361 file path=usr/include/sys/sad.h
1362 $(i386_ONLY)file path=usr/include/sys/sata/sata_defs.h
1363 $(i386_ONLY)file path=usr/include/sys/sata/sata_hba.h
1364 file path=usr/include/sys/schedctl.h
1365 $(sparc_ONLY)file path=usr/include/sys/scsi/adapters/ifpio.h
1366 file path=usr/include/sys/scsi/adapters/scsi_vhci.h
1367 $(sparc_ONLY)file path=usr/include/sys/scsi/adapters/sfvar.h
1368 file path=usr/include/sys/scsi/conf/autoconf.h
1369 file path=usr/include/sys/scsi/conf/device.h
1370 file path=usr/include/sys/scsi/generic/commands.h
1371 file path=usr/include/sys/scsi/generic/dad_mode.h
1372 file path=usr/include/sys/scsi/generic/inquiry.h
1373 file path=usr/include/sys/scsi/generic/message.h
1374 file path=usr/include/sys/scsi/generic/mode.h
1375 file path=usr/include/sys/scsi/generic/persist.h
1376 file path=usr/include/sys/scsi/generic/sense.h
1377 file path=usr/include/sys/scsi/generic/sff_frames.h
1378 file path=usr/include/sys/scsi/generic/smp_frames.h
1379 file path=usr/include/sys/scsi/generic/status.h
```



```
1380 file path=usr/include/sys/scsi/impl/commands.h
1381 file path=usr/include/sys/scsi/impl/inquiry.h
1382 file path=usr/include/sys/scsi/impl/mode.h
1383 file path=usr/include/sys/scsi/impl/scsi_reset_notify.h
1384 file path=usr/include/sys/scsi/impl/scsi_sas.h
1385 file path=usr/include/sys/scsi/impl/sense.h
1386 file path=usr/include/sys/scsi/impl/services.h
1387 file path=usr/include/sys/scsi/impl/smp_transport.h
1388 file path=usr/include/sys/scsi/impl/spc3_types.h
1389 file path=usr/include/sys/scsi/impl/status.h
1390 file path=usr/include/sys/scsi/impl/transport.h
1391 file path=usr/include/sys/scsi/impl/types.h
1392 file path=usr/include/sys/scsi/impl/uscsi.h
1393 file path=usr/include/sys/scsi/impl/usmp.h
1394 file path=usr/include/sys/scsi/scsi.h
1395 file path=usr/include/sys/scsi/scsi_address.h
1396 file path=usr/include/sys/scsi/scsi_ctl.h
1397 file path=usr/include/sys/scsi/scsi_fm.h
1398 file path=usr/include/sys/scsi/scsi_params.h
1399 file path=usr/include/sys/scsi/scsi_pkt.h
1400 file path=usr/include/sys/scsi/scsi_resource.h
1401 file path=usr/include/sys/scsi/scsi_types.h
1402 file path=usr/include/sys/scsi/scsi_watch.h
1403 file path=usr/include/sys/scsi/targets/sddef.h
1404 file path=usr/include/sys/scsi/targets/ses.h
1405 file path=usr/include/sys/scsi/targets/sesio.h
1406 file path=usr/include/sys/scsi/targets/sgendef.h
1407 file path=usr/include/sys/scsi/targets/smp.h
1408 $(sparc_ONLY)file path=usr/include/sys/scsi/targets/ssddef.h
1409 file path=usr/include/sys/scsi/targets/stdef.h
1410 $(i386_ONLY)file path=usr/include/sys/segment.h
1411 $(i386_ONLY)file path=usr/include/sys/segments.h
1412 file path=usr/include/sys/select.h
1413 file path=usr/include/sys/sem.h
1414 file path=usr/include/sys/sem_impl.h
1415 file path=usr/include/sys/sema_impl.h
1416 file path=usr/include/sys/semaphore.h
1417 file path=usr/include/sys/sendfile.h
1418 $(sparc_ONLY)file path=usr/include/sys/ser_async.h
1419 file path=usr/include/sys/ser_sync.h
1420 $(sparc_ONLY)file path=usr/include/sys/ser_zscc.h
1421 file path=usr/include/sys/serializer.h
1422 file path=usr/include/sys/session.h
1423 file path=usr/include/sys/sha1.h
1424 file path=usr/include/sys/sha2.h
1425 file path=usr/include/sys/share.h
1426 file path=usr/include/sys/shm.h
1427 file path=usr/include/sys/shm_impl.h
1428 file path=usr/include/sys/sid.h
1429 file path=usr/include/sys/siginfo.h
1430 file path=usr/include/sys/signal.h
1431 file path=usr/include/sys/sleepq.h
1432 file path=usr/include/sys/smbios.h
1433 file path=usr/include/sys/smbios_impl.h
1434 file path=usr/include/sys/smedia.h
1435 file path=usr/include/sys/sobject.h
1436 $(sparc_ONLY)file path=usr/include/sys/socal_cq_defs.h
1437 $(sparc_ONLY)file path=usr/include/sys/socalio.h
1438 $(sparc_ONLY)file path=usr/include/sys/socalmap.h
1439 $(sparc_ONLY)file path=usr/include/sys/socalreg.h
1440 $(sparc_ONLY)file path=usr/include/sys/socalvar.h
1441 file path=usr/include/sys/socket.h
1442 file path=usr/include/sys/socket_impl.h
1443 file path=usr/include/sys/socket_proto.h
1444 file path=usr/include/sys/socketvar.h
1445 file path=usr/include/sys/sockio.h
```

```
1446 file path=usr/include/sys/spl.h
1447 file path=usr/include/sys/squeue.h
1448 file path=usr/include/sys/squeue_impl.h
1449 file path=usr/include/sys/sservice.h
1450 file path=usr/include/sys/stack.h
1451 file path=usr/include/sys/stat.h
1452 file path=usr/include/sys/stat_impl.h
1453 file path=usr/include/sys/statfs.h
1454 file path=usr/include/sys/statvfs.h
1455 file path=usr/include/sys/stdbool.h
1456 file path=usr/include/sys/stdint.h
1457 file path=usr/include/sys/stermio.h
1458 file path=usr/include/sys/stream.h
1459 file path=usr/include/sys/strft.h
1460 file path=usr/include/sys/strlog.h
1461 file path=usr/include/sys/strmdep.h
1462 file path=usr/include/sys/stropts.h
1463 file path=usr/include/sys/strredir.h
1464 file path=usr/include/sys/strstat.h
1465 file path=usr/include/sys/strsubr.h
1466 file path=usr/include/sys/strsun.h
1467 file path=usr/include/sys/strtty.h
1468 file path=usr/include/sys/sunddi.h
1469 file path=usr/include/sys/sunldi.h
1470 file path=usr/include/sys/sunldi_impl.h
1471 file path=usr/include/sys/sunmdi.h
1472 file path=usr/include/sys/sunndi.h
1473 file path=usr/include/sys/sunpm.h
1474 file path=usr/include/sys/suntpi.h
1475 file path=usr/include/sys/suntty.h
1476 file path=usr/include/sys/swap.h
1477 file path=usr/include/sys/synch.h
1478 file path=usr/include/sys/syscall.h
1479 file path=usr/include/sys/sysconf.h
1480 file path=usr/include/sys/sysconfig.h
1481 file path=usr/include/sys/sysconfig_impl.h
1482 file path=usr/include/sys/sysdc.h
1483 file path=usr/include/sys/sysdc_impl.h
1484 file path=usr/include/sys/sysevent.h
1485 file path=usr/include/sys/sysevent/ap_driver.h
1486 file path=usr/include/sys/sysevent/dev.h
1487 file path=usr/include/sys/sysevent/domain.h
1488 file path=usr/include/sys/sysevent/dr.h
1489 file path=usr/include/sys/sysevent/env.h
1490 file path=usr/include/sys/sysevent/eventdefs.h
1491 file path=usr/include/sys/sysevent/ipmp.h
1492 file path=usr/include/sys/sysevent/pwrctl.h
1493 file path=usr/include/sys/sysevent/svm.h
1494 file path=usr/include/sys/sysevent/vrrp.h
1495 file path=usr/include/sys/sysevent_impl.h
1496 $(i386_ONLY)file path=usr/include/sys/sysi86.h
1497 file path=usr/include/sys/sysinfo.h
1498 file path=usr/include/sys/syslog.h
1499 file path=usr/include/sys/sysmacros.h
1500 file path=usr/include/sys/systeminfo.h
1501 file path=usr/include/sys/systm.h
1502 file path=usr/include/sys/t_kuser.h
1503 file path=usr/include/sys/t_lock.h
1504 file path=usr/include/sys/task.h
1505 file path=usr/include/sys/taskq.h
1506 file path=usr/include/sys/taskq_impl.h
1507 file path=usr/include/sys/telioctl.h
1508 file path=usr/include/sys/termio.h
1509 file path=usr/include/sys/termios.h
1510 file path=usr/include/sys/termiox.h
1511 file path=usr/include/sys/thread.h
```



```
1512 file path=usr/include/sys/ticlts.h
1513 file path=usr/include/sys/ticots.h
1514 file path=usr/include/sys/ticotsord.h
1515 file path=usr/include/sys/tihdr.h
1516 file path=usr/include/sys/time.h
1517 file path=usr/include/sys/time_impl.h
1518 file path=usr/include/sys/time_std_impl.h
1519 file path=usr/include/sys/timeb.h
1520 file path=usr/include/sys/timer.h
1521 file path=usr/include/sys/times.h
1522 file path=usr/include/sys/timex.h
1523 file path=usr/include/sys/timod.h
1524 file path=usr/include/sys/tirdwr.h
1525 file path=usr/include/sys/tiuser.h
1526 file path=usr/include/sys/tl.h
1527 file path=usr/include/sys/tnf.h
1528 file path=usr/include/sys/tnf_com.h
1529 file path=usr/include/sys/tnf_probe.h
1530 file path=usr/include/sys/tnf_writer.h
1531 file path=usr/include/sys/todio.h
1532 file path=usr/include/sys/tpicommon.h
1533 file path=usr/include/sys/trap.h
1534 $(i386_ONLY)file path=usr/include/sys/traptrace.h
1535 file path=usr/include/sys/ts.h
1536 file path=usr/include/sys/tsol/label.h
1537 file path=usr/include/sys/tsol/label_macro.h
1538 file path=usr/include/sys/tsol/priv.h
1539 file path=usr/include/sys/tsol/tndb.h
1540 file path=usr/include/sys/tsol/tsyscall.h
1541 file path=usr/include/sys/tspriocntl.h
1542 $(i386_ONLY)file path=usr/include/sys/tss.h
1543 file path=usr/include/sys/ttcompat.h
1544 file path=usr/include/sys/ttold.h
1545 file path=usr/include/sys/tty.h
1546 file path=usr/include/sys/ttychars.h
1547 file path=usr/include/sys/ttydev.h
1548 $(sparc_ONLY)file path=usr/include/sys/ttymux.h
1549 $(sparc_ONLY)file path=usr/include/sys/ttymuxuser.h
1550 file path=usr/include/sys/tuneable.h
1551 file path=usr/include/sys/turnstile.h
1552 file path=usr/include/sys/types.h
1553 file path=usr/include/sys/types32.h
1554 file path=usr/include/sys/tzfile.h
1555 file path=usr/include/sys/u8_textprep.h
1556 file path=usr/include/sys/uadmin.h
1557 $(i386_ONLY)file path=usr/include/sys/ucode.h
1558 file path=usr/include/sys/ucontext.h
1559 file path=usr/include/sys/uio.h
1560 file path=usr/include/sys/ulimit.h
1561 file path=usr/include/sys/un.h
1562 file path=usr/include/sys/unistd.h
1563 file path=usr/include/sys/user.h
1564 file path=usr/include/sys/ustat.h
1565 file path=usr/include/sys/utime.h
1566 file path=usr/include/sys/utrap.h
1567 file path=usr/include/sys/utsname.h
1568 file path=usr/include/sys/utssys.h
1569 file path=usr/include/sys/uuid.h
1570 file path=usr/include/sys/va_impl.h
1571 file path=usr/include/sys/va_list.h
1572 file path=usr/include/sys/var.h
1573 file path=usr/include/sys/varargs.h
1574 file path=usr/include/sys/vfs.h
1575 file path=usr/include/sys/vfs_opreg.h
1576 file path=usr/include/sys/vfstab.h
1577 file path=usr/include/sys/videodev2.h
```

```
1578 file path=usr/include/sys/visual_io.h
1579 file path=usr/include/sys/vm.h
1580 file path=usr/include/sys/vm_usage.h
1581 file path=usr/include/sys/vmem.h
1582 file path=usr/include/sys/vmem_impl.h
1583 file path=usr/include/sys/vmem_impl_user.h
1584 file path=usr/include/sys/vmparam.h
1585 file path=usr/include/sys/vmsystm.h
1586 file path=usr/include/sys/vnode.h
1587 file path=usr/include/sys/vt.h
1588 file path=usr/include/sys/vtdaemon.h
1589 file path=usr/include/sys/vtoc.h
1590 file path=usr/include/sys/vtrace.h
1591 file path=usr/include/sys/vuid_event.h
1592 file path=usr/include/sys/vuid_queue.h
1593 file path=usr/include/sys/vuid_state.h
1594 file path=usr/include/sys/vuid_store.h
1595 file path=usr/include/sys/vuid_wheel.h
1596 file path=usr/include/sys/wait.h
1597 file path=usr/include/sys/waitq.h
1598 file path=usr/include/sys/watchpoint.h
1599 $(i386_ONLY)file path=usr/include/sys/x86_archext.h
1600 $(i386_ONLY)file path=usr/include/sys/xen_errno.h
1601 file path=usr/include/sys/xti_inet.h
1602 file path=usr/include/sys/xti_osi.h
1603 file path=usr/include/sys/xti_xtiopt.h
1604 file path=usr/include/sys/zcons.h
1605 file path=usr/include/sys/zmod.h
1606 file path=usr/include/sys/zone.h
1607 $(sparc_ONLY)file path=usr/include/sys/zsdev.h
1608 file path=usr/include/sysexits.h
1609 file path=usr/include/syslog.h
1610 file path=usr/include/tar.h
1611 file path=usr/include/tcpd.h
1612 file path=usr/include/term.h
1613 file path=usr/include/termcap.h
1614 file path=usr/include/termio.h
1615 file path=usr/include/termios.h
1616 file path=usr/include/thread.h
1617 file path=usr/include/thread_db.h
1618 file path=usr/include/time.h
1619 file path=usr/include/tiuser.h
1620 file path=usr/include/tsol/label.h
1621 file path=usr/include/tzfile.h
1622 file path=usr/include/ucontext.h
1623 file path=usr/include/ucred.h
1624 file path=usr/include/uid_stp.h
1625 file path=usr/include/ulimit.h
1626 file path=usr/include/umem.h
1627 file path=usr/include/umem_impl.h
1628 file path=usr/include/unctrl.h
1629 file path=usr/include/unistd.h
1630 file path=usr/include/user_attr.h
1631 file path=usr/include/userdefs.h
1632 file path=usr/include/ustat.h
1633 file path=usr/include/utility.h
1634 file path=usr/include/utime.h
1635 file path=usr/include/utmp.h
1636 file path=usr/include/utmpx.h
1637 file path=usr/include/uuid/uuid.h
1638 $(sparc_ONLY)file path=usr/include/v7/sys/machpcb.h
1639 $(sparc_ONLY)file path=usr/include/v7/sys/machtrap.h
1640 $(sparc_ONLY)file path=usr/include/v7/sys/mutex_impl.h
1641 $(sparc_ONLY)file path=usr/include/v7/sys/privregs.h
1642 $(sparc_ONLY)file path=usr/include/v7/sys/prom_isa.h
1643 $(sparc_ONLY)file path=usr/include/v7/sys/psr.h
```



```
1644 $(sparc_ONLY)file path=usr/include/v7/sys/traptrace.h
1645 $(sparc_ONLY)file path=usr/include/v9/sys/asi.h
1646 $(sparc_ONLY)file path=usr/include/v9/sys/machpcb.h
1647 $(sparc_ONLY)file path=usr/include/v9/sys/machtrap.h
1648 $(sparc_ONLY)file path=usr/include/v9/sys/membar.h
1649 $(sparc_ONLY)file path=usr/include/v9/sys/mutex_impl.h
1650 $(sparc_ONLY)file path=usr/include/v9/sys/privregs.h
1651 $(sparc_ONLY)file path=usr/include/v9/sys/prom_isa.h
1652 $(sparc_ONLY)file path=usr/include/v9/sys/psr_compat.h
1653 $(sparc_ONLY)file path=usr/include/v9/sys/vis_simulator.h
1654 file path=usr/include/valtools.h
1655 file path=usr/include/values.h
1656 file path=usr/include/varargs.h
1657 file path=usr/include/vm/anon.h
1658 file path=usr/include/vm/as.h
1659 file path=usr/include/vm/faultcode.h
1660 file path=usr/include/vm/hat.h
1661 file path=usr/include/vm/kpm.h
1662 file path=usr/include/vm/page.h
1663 file path=usr/include/vm/pvn.h
1664 file path=usr/include/vm/rm.h
1665 file path=usr/include/vm/seg.h
1666 file path=usr/include/vm/seg_dev.h
1667 file path=usr/include/vm/seg_enum.h
1668 file path=usr/include/vm/seg_kmem.h
1669 file path=usr/include/vm/seg_kp.h
1670 file path=usr/include/vm/seg_kpm.h
1671 file path=usr/include/vm/seg_map.h
1672 file path=usr/include/vm/seg_spt.h
1673 file path=usr/include/vm/seg_vn.h
1674 file path=usr/include/vm/vpage.h
1675 file path=usr/include/vm/vpm.h
1676 file path=usr/include/volmgt.h
1677 file path=usr/include/wait.h
1678 file path=usr/include/wchar.h
1679 file path=usr/include/wchar_impl.h
1680 file path=usr/include/wctype.h
1681 file path=usr/include/widec.h
1682 file path=usr/include/wordexp.h
1683 file path=usr/include/xlocale.h
1684 file path=usr/include/xti.h
1685 file path=usr/include/xti_inet.h
1686 file path=usr/include/zone.h
1687 file path=usr/include/zonestat.h
1688 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/acpidev.h
1689 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/amd_iommu.h
1690 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/asm_misc.h
1691 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/clock.h
1692 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/cram.h
1693 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/ddi_subrdefs.h
1694 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/debug_info.h
1695 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/fastboot.h
1696 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/mach_mmu.h
1697 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machclock.h
1698 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machcpuvar.h
1699 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machparam.h
1700 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machprivregs.h
1701 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machsystm.h
1702 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machthread.h
1703 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/memnode.h
1704 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/pc_mmu.h
1705 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/psm.h
1706 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/psm_defs.h
1707 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/psm_modctl.h
1708 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/psm_types.h
1709 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/rm_platter.h
```

```
1710 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/sbd_ioctl.h
1711 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/smp_impldefs.h
1712 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/vm_machparam.h
1713 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/x_call.h
1714 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/xc_levels.h
1715 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/xsvc.h
1716 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/hat_i86.h
1717 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/hat_pte.h
1718 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/hment.h
1719 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/htable.h
1720 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/kboot_mmu.h
1721 $(i386_ONLY)file path=usr/platform/i86xpv/include/sys/balloon.h
1722 $(i386_ONLY)file path=usr/platform/i86xpv/include/sys/machprivregs.h
1723 $(i386_ONLY)file path=usr/platform/i86xpv/include/sys/xen_mmu.h
1724 $(i386_ONLY)file path=usr/platform/i86xpv/include/sys/xpv_impl.h
1725 $(i386_ONLY)file path=usr/platform/i86xpv/include/vm/seg_mf.h
1726 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/ac.h
1727 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/async.h
1728 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cheetahregs.h
1729 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cherrystone.h
1730 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/clock.h
1731 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cmp.h
1732 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cpc_ultra.h
1733 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cpr_impl.h
1734 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cpu_impl.h
1735 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cpu_sgnblk_defs.h
1736 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cvc.h
1737 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/daktari.h
1738 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/ddi_subrdefs.h
1739 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/dvma.h
1740 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/ecc_kstat.h
1741 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/eeprom.h
1742 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/envctrl.h
1743 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/envctrl_gen.h
1744 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/envctrl_ue250.h
1745 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/envctrl_ue450.h
1746 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/environ.h
1747 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/errclassify.h
1748 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/fhc.h
1749 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/gpio_87317.h
1750 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/hpc3130_events.h
1751 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/clients/hpc3130.h
1752 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/clients/i2c_client.h
1753 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/clients/lm75.h
1754 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/clients/max1617.h
1755 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/clients/pcf8591.h
1756 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/clients/ssc050.h
1757 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/misc/i2c_svc.h
1758 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/idprom.h
1759 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/intr.h
1760 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/intreg.h
1761 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/iocache.h
1762 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/iommu.h
1763 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/ivintr.h
1764 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/lom_io.h
1765 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machasi.h
1766 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machclock.h
1767 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machcpuvar.h
1768 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machparam.h
1769 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machsystm.h
1770 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machthread.h
1771 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/mem_cache.h
1772 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/memlist_plat.h
1773 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/memnode.h
1774 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/mmu.h
1775 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/nexusdebug.h
```



```
1776 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/opl_hwdesc.h
1777 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/opl_module.h
1778 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/prom_debug.h
1779 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/prom_plat.h
1780 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/pte.h
1781 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sbd_ioctl.h
1782 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/scb.h
1783 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/scsb_led.h
1784 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/simmstat.h
1785 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/spitregs.h
1786 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sram.h
1787 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/starfire.h
1788 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sun4asi.h
1789 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sysctrl.h
1790 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sysioerr.h
1791 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sysiosbus.h
1792 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/tod.h
1793 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/todmostek.h
1794 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/trapstat.h
1795 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/traptrace.h
1796 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/vis.h
1797 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/vm_machparam.h
1798 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/x_call.h
1799 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/xc_impl.h
1800 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/zsmach.h
1801 $(sparc_ONLY)file path=usr/platform/sun4u/include/vm/hat_sfmmu.h
1802 $(sparc_ONLY)file path=usr/platform/sun4u/include/vm/mach_sfmmu.h
1803 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/clock.h
1804 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/cmp.h
1805 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/cpc_ultra.h
1806 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/cpu_sgnblk_defs.h
1807 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ddi_subrdefs.h
1808 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ds_pri.h
1809 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ds_snmp.h
1810 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/dvma.h
1811 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/eeprom.h
1812 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/fcode.h
1813 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/hsvc.h
1814 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/hypervisor_api.h
1815 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/idprom.h
1816 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/intr.h
1817 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/intreg.h
1818 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ivintr.h
1819 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machasi.h
1820 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machclock.h
1821 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machcpuvar.h
1822 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machintreg.h
1823 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machparam.h
1824 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machsystm.h
1825 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machthread.h
1826 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/memlist_plat.h
1827 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/memnode.h
1828 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/mmu.h
1829 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/nexusdebug.h
1830 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/niagaraasi.h
1831 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/niagararegs.h
1832 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ntwdt.h
1833 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/pri.h
1834 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/prom_debug.h
1835 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/prom_plat.h
1836 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/pte.h
1837 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/qcn.h
1838 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/scb.h
1839 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/soft_state.h
1840 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/sun4asi.h
1841 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/tod.h
```

```
1842 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/trapstat.h
1843 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/traptrace.h
1844 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/vis.h
1845 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/vm_machparam.h
1846 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/x_call.h
1847 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/xc_impl.h
1848 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/zsmach.h
1849 $(sparc_ONLY)file path=usr/platform/sun4v/include/vm/hat_sfmmu.h
1850 $(sparc_ONLY)file path=usr/platform/sun4v/include/vm/mach_sfmmu.h
1851 file path=usr/share/man/man3head/acct.h.3head
1852 file path=usr/share/man/man3head/aio.h.3head
1853 file path=usr/share/man/man3head/ar.h.3head
1854 file path=usr/share/man/man3head/archives.h.3head
1855 file path=usr/share/man/man3head/assert.h.3head
1856 file path=usr/share/man/man3head/complex.h.3head
1857 file path=usr/share/man/man3head/cpio.h.3head
1858 file path=usr/share/man/man3head/dirent.h.3head
1859 file path=usr/share/man/man3head/errno.h.3head
1860 file path=usr/share/man/man3head/fcntl.h.3head
1861 file path=usr/share/man/man3head/fenv.h.3head
1862 file path=usr/share/man/man3head/float.h.3head
1863 file path=usr/share/man/man3head/floatingpoint.h.3head
1864 file path=usr/share/man/man3head/fmtmsg.h.3head
1865 file path=usr/share/man/man3head/fnmatch.h.3head
1866 file path=usr/share/man/man3head/ftw.h.3head
1867 file path=usr/share/man/man3head/glob.h.3head
1868 file path=usr/share/man/man3head/grp.h.3head
1869 file path=usr/share/man/man3head/iconv.h.3head
1870 file path=usr/share/man/man3head/if.h.3head
1871 file path=usr/share/man/man3head/in.h.3head
1872 file path=usr/share/man/man3head/inet.h.3head
1873 file path=usr/share/man/man3head/inttypes.h.3head
1874 file path=usr/share/man/man3head/ipc.h.3head
1875 file path=usr/share/man/man3head/iso646.h.3head
1876 file path=usr/share/man/man3head/langinfo.h.3head
1877 file path=usr/share/man/man3head/libgen.h.3head
1878 file path=usr/share/man/man3head/libintl.h.3head
1879 file path=usr/share/man/man3head/limits.h.3head
1880 file path=usr/share/man/man3head/locale.h.3head
1881 file path=usr/share/man/man3head/math.h.3head
1882 file path=usr/share/man/man3head/mman.h.3head
1883 file path=usr/share/man/man3head/monetary.h.3head
1884 file path=usr/share/man/man3head/mqueue.h.3head
1885 file path=usr/share/man/man3head/msg.h.3head
1886 file path=usr/share/man/man3head/ndbm.h.3head
1887 file path=usr/share/man/man3head/netdb.h.3head
1888 file path=usr/share/man/man3head/nl_types.h.3head
1889 file path=usr/share/man/man3head/poll.h.3head
1890 file path=usr/share/man/man3head/pthread.h.3head
1891 file path=usr/share/man/man3head/pwd.h.3head
1892 file path=usr/share/man/man3head/regex.h.3head
1893 file path=usr/share/man/man3head/resource.h.3head
1894 file path=usr/share/man/man3head/sched.h.3head
1895 file path=usr/share/man/man3head/search.h.3head
1896 file path=usr/share/man/man3head/select.h.3head
1897 file path=usr/share/man/man3head/sem.h.3head
1898 file path=usr/share/man/man3head/semaphore.h.3head
1899 file path=usr/share/man/man3head/setjmp.h.3head
1900 file path=usr/share/man/man3head/shm.h.3head
1901 file path=usr/share/man/man3head/siginfo.h.3head
1902 file path=usr/share/man/man3head/signal.h.3head
1903 file path=usr/share/man/man3head/socket.h.3head
1904 file path=usr/share/man/man3head/spawn.h.3head
1905 file path=usr/share/man/man3head/stat.h.3head
1906 file path=usr/share/man/man3head/statvfs.h.3head
1907 file path=usr/share/man/man3head/stdbool.h.3head
```



```
1908 file path=usr/share/man/man3head/stddef.h.3head
1909 file path=usr/share/man/man3head/stdint.h.3head
1910 file path=usr/share/man/man3head/stdio.h.3head
1911 file path=usr/share/man/man3head/stdlib.h.3head
1912 file path=usr/share/man/man3head/string.h.3head
1913 file path=usr/share/man/man3head/strings.h.3head
1914 file path=usr/share/man/man3head/stropts.h.3head
1915 file path=usr/share/man/man3head/syslog.h.3head
1916 file path=usr/share/man/man3head/tar.h.3head
1917 file path=usr/share/man/man3head/tcp.h.3head
1918 file path=usr/share/man/man3head/termios.h.3head
1919 file path=usr/share/man/man3head/tgmath.h.3head
1920 file path=usr/share/man/man3head/time.h.3head
1921 file path=usr/share/man/man3head/timeb.h.3head
1922 file path=usr/share/man/man3head/times.h.3head
1923 file path=usr/share/man/man3head/types.h.3head
1924 file path=usr/share/man/man3head/types32.h.3head
1925 file path=usr/share/man/man3head/ucontext.h.3head
1926 file path=usr/share/man/man3head/uio.h.3head
1927 file path=usr/share/man/man3head/ulimit.h.3head
1928 file path=usr/share/man/man3head/un.h.3head
1929 file path=usr/share/man/man3head/unistd.h.3head
1930 file path=usr/share/man/man3head/utime.h.3head
1931 file path=usr/share/man/man3head/utmpx.h.3head
1932 file path=usr/share/man/man3head/utsname.h.3head
1933 file path=usr/share/man/man3head/values.h.3head
1934 file path=usr/share/man/man3head/wait.h.3head
1935 file path=usr/share/man/man3head/wchar.h.3head
1936 file path=usr/share/man/man3head/wctype.h.3head
1937 file path=usr/share/man/man3head/wordexp.h.3head
1938 file path=usr/share/man/man3head/xlocale.h.3head
1939 file path=usr/share/man/man4/note.4
1940 file path=usr/share/man/man5/prof.5
1941 file path=usr/share/man/man7i/cdio.7i
1942 file path=usr/share/man/man7i/dkio.7i
1943 file path=usr/share/man/man7i/fbio.7i
1944 file path=usr/share/man/man7i/fdio.7i
1945 file path=usr/share/man/man7i/hdio.7i
1946 file path=usr/share/man/man7i/iec61883.7i
1947 file path=usr/share/man/man7i/mhd.7i
1948 file path=usr/share/man/man7i/mtio.7i
1949 file path=usr/share/man/man7i/prnio.7i
1950 file path=usr/share/man/man7i/quotactl.7i
1951 file path=usr/share/man/man7i/sesio.7i
1952 file path=usr/share/man/man7i/sockio.7i
1953 file path=usr/share/man/man7i/streamio.7i
1954 file path=usr/share/man/man7i/termio.7i
1955 file path=usr/share/man/man7i/termiox.7i
1956 file path=usr/share/man/man7i/uscsi.7i
1957 file path=usr/share/man/man7i/visual_io.7i
1958 file path=usr/share/man/man7i/vt.7i
1959 file path=usr/xpg4/include/curses.h
1960 file path=usr/xpg4/include/term.h
1961 file path=usr/xpg4/include/unctrl.h
1962 legacy pkg=SUNWhea \
1963     desc="SunOS C/C++ header files for general development of software" \
1964     name="SunOS Header Files"
1965 license cr_Sun license=cr_Sun
1966 license lic_CDDL license=lic_CDDL
1967 license license_in_headers license=license_in_headers
1968 license usr/src/lib/pkcs11/include/THIRDPARTYLICENSE \
1969     license=usr/src/lib/pkcs11/include/THIRDPARTYLICENSE
1970 link path=usr/include/iso/assert_iso.h target=../assert.h
1971 link path=usr/include/iso/errno_iso.h target=../errno.h
1972 link path=usr/include/iso/float_iso.h target=../float.h
1973 link path=usr/include/iso/iso646_iso.h target=../iso646.h
```

```
$(sparc_ONLY)link path=usr/platform/SUNW,A70/include target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Netra-T12/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Netra-T4/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,SPARC-Enterprise/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Serverblade1/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Blade-100/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Blade-1000/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Blade-1500/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Blade-2500/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-15000/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-280R/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-480R/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-880/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V215/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V240/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V250/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V440/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V445/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V490/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V890/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Ultra-2/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Ultra-250/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Ultra-4/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Ultra-Enterprise-10000/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,Ultra-Enterprise/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,UltraSPARC-IIe-NetraCT-40/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,UltraSPARC-IIe-NetraCT-60/include \
    target=../sun4u/include
$(sparc_ONLY)link path=usr/platform/SUNW,UltraSPARC-IIi-Netract/include \
    target=../sun4u/include
link path=usr/share/man/man3head/acct.3head target=acct.h.3head
link path=usr/share/man/man3head/aio.3head target=aio.h.3head
link path=usr/share/man/man3head/ar.3head target=ar.h.3head
link path=usr/share/man/man3head/archives.3head target=archives.h.3head
link path=usr/share/man/man3head/assert.3head target=assert.h.3head
link path=usr/share/man/man3head/complex.3head target=complex.h.3head
link path=usr/share/man/man3head/cpio.3head target=cpio.h.3head
link path=usr/share/man/man3head/dirent.3head target=dirent.h.3head
link path=usr/share/man/man3head/errno.3head target=errno.h.3head
```



```
link path=usr/share/man/man3head/fcntl.3head target=fcntl.h.3head
link path=usr/share/man/man3head/fenv.3head target=fenv.h.3head
link path=usr/share/man/man3head/float.3head target=float.h.3head
link path=usr/share/man/man3head/floatingpoint.3head \
    target=floatingpoint.h.3head
link path=usr/share/man/man3head/fmtmsg.3head target=fmtmsg.h.3head
link path=usr/share/man/man3head/fnmatch.3head target=fnmatch.h.3head
link path=usr/share/man/man3head/ftw.3head target=ftw.h.3head
link path=usr/share/man/man3head/glob.3head target=glob.h.3head
link path=usr/share/man/man3head/grp.3head target=grp.h.3head
link path=usr/share/man/man3head/iconv.3head target=iconv.h.3head
link path=usr/share/man/man3head/if.3head target=if.h.3head
link path=usr/share/man/man3head/in.3head target=in.h.3head
link path=usr/share/man/man3head/inet.3head target=inet.h.3head
link path=usr/share/man/man3head/inttypes.3head target=inttypes.h.3head
link path=usr/share/man/man3head/ipc.3head target=ipc.h.3head
link path=usr/share/man/man3head/iso646.3head target=iso646.h.3head
link path=usr/share/man/man3head/langinfo.3head target=langinfo.h.3head
link path=usr/share/man/man3head/libgen.3head target=libgen.h.3head
link path=usr/share/man/man3head/libintl.3head target=libintl.h.3head
link path=usr/share/man/man3head/limits.3head target=limits.h.3head
link path=usr/share/man/man3head/locale.3head target=locale.h.3head
link path=usr/share/man/man3head/math.3head target=math.h.3head
link path=usr/share/man/man3head/mman.3head target=mman.h.3head
link path=usr/share/man/man3head/monetary.3head target=monetary.h.3head
link path=usr/share/man/man3head/mqueue.3head target=mqueue.h.3head
link path=usr/share/man/man3head/msg.3head target=msg.h.3head
link path=usr/share/man/man3head/ndbm.3head target=ndbm.h.3head
link path=usr/share/man/man3head/netdb.3head target=netdb.h.3head
link path=usr/share/man/man3head/nl_types.3head target=nl_types.h.3head
link path=usr/share/man/man3head/poll.3head target=poll.h.3head
link path=usr/share/man/man3head/pthread.3head target=pthread.h.3head
link path=usr/share/man/man3head/pwd.3head target=pwd.h.3head
link path=usr/share/man/man3head/regex.3head target=regex.h.3head
link path=usr/share/man/man3head/resource.3head target=resource.h.3head
link path=usr/share/man/man3head/sched.3head target=sched.h.3head
link path=usr/share/man/man3head/search.3head target=search.h.3head
link path=usr/share/man/man3head/select.3head target=select.h.3head
link path=usr/share/man/man3head/sem.3head target=sem.h.3head
link path=usr/share/man/man3head/semaphore.3head target=semaphore.h.3head
link path=usr/share/man/man3head/setjmp.3head target=setjmp.h.3head
link path=usr/share/man/man3head/shm.3head target=shm.h.3head
link path=usr/share/man/man3head/siginfo.3head target=siginfo.h.3head
link path=usr/share/man/man3head/signal.3head target=signal.h.3head
link path=usr/share/man/man3head/socket.3head target=socket.h.3head
link path=usr/share/man/man3head/spawn.3head target=spawn.h.3head
link path=usr/share/man/man3head/stat.3head target=stat.h.3head
link path=usr/share/man/man3head/statvfs.3head target=statvfs.h.3head
link path=usr/share/man/man3head/stdbool.3head target=stdbool.h.3head
link path=usr/share/man/man3head/stddef.3head target=stddef.h.3head
link path=usr/share/man/man3head/stdint.3head target=stdint.h.3head
link path=usr/share/man/man3head/stdio.3head target=stdio.h.3head
link path=usr/share/man/man3head/stdlib.3head target=stdlib.h.3head
link path=usr/share/man/man3head/string.3head target=string.h.3head
link path=usr/share/man/man3head/strings.3head target=strings.h.3head
link path=usr/share/man/man3head/stropts.3head target=stropts.h.3head
link path=usr/share/man/man3head/syslog.3head target=syslog.h.3head
link path=usr/share/man/man3head/tar.3head target=tar.h.3head
link path=usr/share/man/man3head/tcp.3head target=tcp.h.3head
link path=usr/share/man/man3head/termios.3head target=termios.h.3head
link path=usr/share/man/man3head/tgmath.3head target=tgmath.h.3head
link path=usr/share/man/man3head/time.3head target=time.h.3head
link path=usr/share/man/man3head/timeb.3head target=timeb.h.3head
link path=usr/share/man/man3head/times.3head target=times.h.3head
link path=usr/share/man/man3head/types.3head target=types.h.3head
link path=usr/share/man/man3head/types32.3head target=types32.h.3head
```

```
2106 link path=usr/share/man/man3head/ucontext.3head target=ucontext.h.3head
2107 link path=usr/share/man/man3head/uio.3head target=uio.h.3head
2108 link path=usr/share/man/man3head/ulimit.3head target=ulimit.h.3head
2109 link path=usr/share/man/man3head/un.3head target=un.h.3head
2110 link path=usr/share/man/man3head/unistd.3head target=unistd.h.3head
2111 link path=usr/share/man/man3head/utime.3head target=utime.h.3head
2112 link path=usr/share/man/man3head/utmpx.3head target=utmpx.h.3head
2113 link path=usr/share/man/man3head/utsname.3head target=utsname.h.3head
2114 link path=usr/share/man/man3head/values.3head target=values.h.3head
2115 link path=usr/share/man/man3head/wait.3head target=wait.h.3head
2116 link path=usr/share/man/man3head/wchar.3head target=wchar.h.3head
2117 link path=usr/share/man/man3head/wctype.3head target=wctype.h.3head
2118 link path=usr/share/man/man3head/wordexp.3head target=wordexp.h.3head
2119 link path=usr/share/man/man3head/xlocale.3head target=xlocale.h.3head
2120 $(i386_ONLY)link path=usr/share/src/uts/i86pc/sys \
2121     target=../../../../platform/i86pc/include/sys
2122 $(i386_ONLY)link path=usr/share/src/uts/i86pc/vm \
2123     target=../../../../platform/i86pc/include/vm
2124 $(i386_ONLY)link path=usr/share/src/uts/i86xpv/sys \
2125     target=../../../../platform/i86xpv/include/sys
2126 $(i386_ONLY)link path=usr/share/src/uts/i86xpv/vm \
2127     target=../../../../platform/i86xpv/include/vm
2128 $(sparc_ONLY)link path=usr/share/src/uts/sun4u/sys \
2129     target=../../../../platform/sun4u/include/sys
2130 $(sparc_ONLY)link path=usr/share/src/uts/sun4u/vm \
2131     target=../../../../platform/sun4u/include/vm
2132 $(sparc_ONLY)link path=usr/share/src/uts/sun4v/sys \
2133     target=../../../../platform/sun4v/include/sys
2134 $(sparc_ONLY)link path=usr/share/src/uts/sun4v/vm \
2135     target=../../../../platform/sun4v/include/vm
```

****
4832 Sun Dec 14 23:31:26 2014
new/usr/src/uts/common/gssapi/mechs/krb5/crypto/cksumtypes.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
****

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


/*
 * Copyright (C) 1998 by the FundsXpress, INC.
 *
 * All rights reserved.
 *
 * Export of this software from the United States of America may require
 * a specific license from the United States Government.  It is the
 * responsibility of any person or organization contemplating export to
 * obtain such a license before exporting.
 *
 * WITHIN THAT CONSTRAINT, permission to use, copy, modify, and
 * distribute this software and its documentation for any purpose and
 * without fee is hereby granted, provided that the above copyright
 * notice appear in all copies and that both that copyright notice and
 * this permission notice appear in supporting documentation, and that
 * the name of FundsXpress. not be used in advertising or publicity pertaining
 * to distribution of the software without specific, written prior
 * permission.  FundsXpress makes no representations about the suitability of
 * this software for any purpose.  It is provided "as is" without express
 * or implied warranty.
 *
 * THIS SOFTWARE IS PROVIDED ''AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.
 */

#include "k5-int.h"
#include "hash_provider.h"
#include "keyhash_provider.h"
#include "cksumtypes.h"

const struct krb5_cksumtypes krb5_cksumtypes_list[] = {
    { CKSUMTYPE_CRC32, KRB5_CKSUMFLAG_NOT_COLL_PROOF,
      "crc32", "CRC-32",
      0, NULL, &krb5int_hash_crc32, 0,
      NULL, NULL, &krb5int_hash_crc32, 0,
#ifdef _KERNEL
      NULL,
      CRYPTO_MECH_INVALID
#endif /* _KERNEL */
},

    { CKSUMTYPE_DESCBC, 0,
      "des-cbc", "DES cbc mode",
      ENCTYPE_DES_CBC_CRC, &krb5int_keyhash_descbc,
      NULL, 0,
      NULL,  NULL,
#ifdef _KERNEL
      NULL,
      CRYPTO_MECH_INVALID
#endif /* _KERNEL */
},
```



```
    { CKSUMTYPE_RSA_MD5, 0,
      "md5", "RSA-MD5",
      0, NULL, &krb5int_hash_md5, 0,
      NULL, NULL, &krb5int_hash_md5, 0,
#ifdef _KERNEL
      SUN_CKM_MD5,
      CRYPTO_MECH_INVALID
#endif /* _KERNEL */
},
    { CKSUMTYPE_RSA_MD5_DES, 0,
      "md5-des", "RSA-MD5 with DES cbc mode",
      ENCTYPE_DES_CBC_CRC, &krb5int_keyhash_md5des,
      NULL, 0,
      NULL, NULL,
#ifdef _KERNEL
      SUN_CKM_MD5,
      CRYPTO_MECH_INVALID
#endif /* _KERNEL */
},

    { CKSUMTYPE_NIST_SHA, 0,
      "sha", "NIST-SHA",
      0, NULL, &krb5int_hash_sha1, 0,
      NULL, NULL, &krb5int_hash_sha1, 0,
#ifdef _KERNEL
      SUN_CKM_SHA1,
      CRYPTO_MECH_INVALID
#endif /* _KERNEL */
},

    { CKSUMTYPE_HMAC_SHA1_DES3, KRB5_CKSUMFLAG_DERIVE,
      "hmac-sha1-des3", "HMAC-SHA1 DES3 key",
      0, NULL, &krb5int_hash_sha1, 0,
      NULL, NULL, &krb5int_hash_sha1, 0,
#ifdef _KERNEL
      SUN_CKM_SHA1_HMAC,
      CRYPTO_MECH_INVALID
#endif /* _KERNEL */
 },
    { CKSUMTYPE_HMAC_SHA1_DES3, KRB5_CKSUMFLAG_DERIVE,
      "hmac-sha1-des3-kd", "HMAC-SHA1 DES3 key", /* alias */
      0, NULL, &krb5int_hash_sha1, 0,
      NULL, NULL, &krb5int_hash_sha1, 0,
#ifdef _KERNEL
      SUN_CKM_SHA1_HMAC,
      CRYPTO_MECH_INVALID
#endif /* _KERNEL */
},
    { CKSUMTYPE_HMAC_MD5_ARCFOUR, 0,
      "hmac-md5-rc4", "Microsoft HMAC MD5 (RC4 key)",
      ENCTYPE_ARCFOUR_HMAC, &krb5int_keyhash_hmac_md5,
      NULL, 0,
#ifdef _KERNEL
      SUN_CKM_MD5,
      CRYPTO_MECH_INVALID
#endif /* _KERNEL */
    },
    { CKSUMTYPE_HMAC_MD5_ARCFOUR, 0,
      "hmac-md5-enc", "Microsoft HMAC MD5 (RC4 key)",  /*Heimdal alias*/
      ENCTYPE_ARCFOUR_HMAC, &krb5int_keyhash_hmac_md5,
      NULL, 0,
#ifdef _KERNEL
      SUN_CKM_MD5,
      CRYPTO_MECH_INVALID
#endif /* _KERNEL */
    },
```

```
119     { CKSUMTYPE_HMAC_MD5_ARCFOUR, 0,
120       "hmac-md5-earcfour", "Microsoft HMAC MD5 (RC4 key)",  /* alias*/
121       ENCTYPE_ARCFOUR_HMAC, &krb5int_keyhash_hmac_md5,
122       NULL, 0,
123 #ifdef _KERNEL
124       SUN_CKM_MD5,
125       CRYPTO_MECH_INVALID
126 #endif /* _KERNEL */
127     },

129     { CKSUMTYPE_HMAC_SHA1_96_AES128, KRB5_CKSUMFLAG_DERIVE,
130       "hmac-sha1-96-aes128", "HMAC-SHA1 AES128 key",
131        0, NULL, &krb5int_hash_sha1, 12,
131        NULL, NULL, &krb5int_hash_sha1, 12,
132 #ifdef _KERNEL
133       SUN_CKM_SHA1_HMAC,
134       CRYPTO_MECH_INVALID
135 #endif /* _KERNEL */
136     },
137     { CKSUMTYPE_HMAC_SHA1_96_AES256, KRB5_CKSUMFLAG_DERIVE,
138       "hmac-sha1-96-aes256", "HMAC-SHA1 AES256 key",
139        0, NULL, &krb5int_hash_sha1, 12,
140 #ifdef _KERNEL
141       SUN_CKM_SHA1_HMAC,
142       CRYPTO_MECH_INVALID
143 #endif /* _KERNEL */
144     }

146 };
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   12447 Sun Dec 14 23:31:27 2014
new/usr/src/uts/common/gssapi/mechs/krb5/crypto/etypes.c
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
______unchanged_portion_omitted_
  87     { ENCTYPE_DES_CBC_MD5,
  88       "des", "DES cbc mode with RSA-MD5", /* alias */
  89       &krb5int_enc_des, &krb5int_hash_md5,
  90       krb5_old_encrypt_length, krb5_old_encrypt, krb5_old_decrypt,
  91       CKSUMTYPE_RSA_MD5,
  92 #ifndef _KERNEL
  93       krb5int_des_string_to_key,
  94 #else
  95       SUN_CKM_DES_CBC,
  96       SUN_CKM_MD5,
  97       CRYPTO_MECH_INVALID,
  98       CRYPTO_MECH_INVALID
  99 #endif /* _KERNEL */
 100  },
 101     { ENCTYPE_DES_CBC_RAW,
 102       "des-cbc-raw", "DES cbc mode raw",
 103       &krb5int_enc_des, NULL,
 104       krb5_raw_encrypt_length, krb5_raw_encrypt, krb5_raw_decrypt, 0,
 104       krb5_raw_encrypt_length, krb5_raw_encrypt, krb5_raw_decrypt,
 105       NULL,
 105 #ifndef _KERNEL
 106       krb5int_des_string_to_key,
 107 #else
 108       SUN_CKM_DES_CBC,
 109       NULL,
 110       CRYPTO_MECH_INVALID,
 111       CRYPTO_MECH_INVALID
 112 #endif /* !_KERNEL */
 113 },

 115     { ENCTYPE_DES3_CBC_RAW,
 116       "des3-cbc-raw", "Triple DES cbc mode raw",
 117       &krb5int_enc_des3, NULL,
 118       krb5_raw_encrypt_length, krb5_raw_encrypt, krb5_raw_decrypt, 0,
 119       krb5_raw_encrypt_length, krb5_raw_encrypt, krb5_raw_decrypt,
 120       NULL,
 119 #ifndef _KERNEL
 120       krb5int_dk_string_to_key,
 121 #else
 122       SUN_CKM_DES3_CBC,
 123       NULL,
 124       CRYPTO_MECH_INVALID,
 125       CRYPTO_MECH_INVALID
 126 #endif /* !_KERNEL */
 127 },
______unchanged_portion_omitted_

 189     { ENCTYPE_DES_HMAC_SHA1,
 190       "des-hmac-sha1", "DES with HMAC/sha1",
 191       &krb5int_enc_des, &krb5int_hash_sha1,
 192       krb5_dk_encrypt_length, krb5_dk_encrypt, krb5_dk_decrypt, 0,
 194       krb5_dk_encrypt_length, krb5_dk_encrypt, krb5_dk_decrypt,
 195       NULL,
 193 #ifndef _KERNEL
 194       krb5int_dk_string_to_key,
 195 #else
 196       SUN_CKM_DES_CBC,
 197       SUN_CKM_SHA1_HMAC,
```



```
 198       CRYPTO_MECH_INVALID,
 199       CRYPTO_MECH_INVALID
 200 #endif /* !_KERNEL */
 201 },
______unchanged_portion_omitted_
```

```
**********************************************************
   22334 Sun Dec 14 23:31:27 2014
new/usr/src/uts/common/sys/Makefile
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2013, Joyent, Inc. All rights reserved.
# Copyright 2013 Garrett D'Amore <garrett@damore.org>
#

include $(SRC)/uts/Makefile.uts

FILEMODE=644

#
#       Note that the following headers are present in the kernel but
#       neither installed or shipped as part of the product:
#               cpuid_drv.h:            Private interface for cpuid consumers
#               unix_bb_info.h:         Private interface to kcov
#

i386_HDRS=                      \
        agp/agpamd64gart_io.h   \
        agp/agpdefs.h           \
        agp/agpgart_impl.h      \
        agp/agpmaster_io.h      \
        agp/agptarget_io.h      \
        agpgart.h               \
        asy.h                   \
        fd_debug.h              \
        fdc.h                   \
        fdmedia.h               \
        mouse.h                 \
        ucode.h

sparc_HDRS=                     \
        mouse.h                 \
        scsi/targets/ssddef.h   \
        $(MDESCHDRS)

# Generated headers
GENHDRS=                        \
        priv_const.h            \
```



```
        priv_names.h            \
        usb/usbdevs.h

CHKHDRS=                        \
        acpi_drv.h              \
        acct.h                  \
        acctctl.h               \
        acl.h                   \
        acl_impl.h              \
        aggr.h                  \
        aggr_impl.h             \
        aio.h                   \
        aio_impl.h              \
        aio_req.h               \
        aiocb.h                 \
        ascii.h                 \
        asynch.h                \
        atomic.h                \
        attr.h                  \
        audio.h                 \
        audioio.h               \
        autoconf.h              \
        auxv.h                  \
        auxv_386.h              \
        auxv_SPARC.h            \
        avl.h                   \
        avl_impl.h              \
        bitmap.h                \
        bitset.h                \
        bl.h                    \
        blkdev.h                \
        bofi.h                  \
        bofi_impl.h             \
        bpp_io.h                \
        bootstat.h              \
        brand.h                 \
        buf.h                   \
        bufmod.h                \
        bustypes.h              \
        byteorder.h             \
        callb.h                 \
        callo.h                 \
        cap_util.h              \
        cpucaps.h               \
        cpucaps_impl.h          \
        ccompile.h              \
        cdio.h                  \
        cladm.h                 \
        class.h                 \
        clconf.h                \
        clock_impl.h            \
        cmlb.h                  \
        cmn_err.h               \
        compress.h              \
        condvar.h               \
        condvar_impl.h          \
        conf.h                  \
        consdev.h               \
        console.h               \
        consplat.h              \
        vt.h                    \
        vtdaemon.h              \
        kd.h                    \
        contract.h              \
        contract_impl.h         \
        copyops.h               \
```

```
126         core.h                  \
127         corectl.h               \
128         cpc_impl.h              \
129         cpc_pcbe.h              \
130         cpr.h                   \
131         cpupart.h               \
132         cpuvar.h                \
133         crc32.h                 \
134         cred.h                  \
135         cred_impl.h             \
136         crtctl.h                \
137         cryptmod.h              \
138         csiioctl.h              \
139         ctf.h                   \
140         ctfs.h                  \
141         ctfs_impl.h             \
142         ctf_api.h               \
143         ctype.h                 \
144         cyclic.h                \
145         cyclic_impl.h           \
146         dacf.h                  \
147         dacf_impl.h             \
148         damap.h                 \
149         damap_impl.h            \
150         dc_ki.h                 \
151         ddi.h                   \
152         ddifm.h                 \
153         ddifm_impl.h            \
154         ddi_hp.h                \
155         ddi_hp_impl.h           \
156         ddi_intr.h              \
157         ddi_intr_impl.h         \
158         ddi_impldefs.h          \
159         ddi_implfuncs.h         \
160         ddi_obsolete.h          \
161         ddi_periodic.h          \
162         ddidevmap.h             \
163         ddidmareq.h             \
164         ddimapreq.h             \
165         ddipropdefs.h           \
166         dditypes.h              \
167         debug.h                 \
168         des.h                   \
169         devctl.h                \
170         devcache.h              \
171         devcache_impl.h         \
172         devfm.h                 \
173         devid_cache.h           \
174         devinfo_impl.h          \
175         devops.h                \
176         devpolicy.h             \
177         devpoll.h               \
178         dirent.h                \
179         disp.h                  \
180         dkbad.h                 \
181         dkio.h                  \
182         dklabel.h               \
183         dl.h                    \
184         dlpi.h                  \
185         dld.h                   \
186         dld_impl.h              \
187         dld_ioc.h               \
188         dls.h                   \
189         dls_mgmt.h              \
190         dls_impl.h              \
191         dma_i8237A.h            \
```



```
192         dnlc.h                  \
193         door.h                  \
194         door_data.h             \
195         door_impl.h             \
196         dtrace.h                \
197         dtrace_impl.h           \
198         dumpadm.h               \
199         dumphdr.h               \
200         ecppsys.h               \
201         ecppio.h                \
202         ecppreg.h               \
203         ecppvar.h               \
204         efi_partition.h         \
205         elf.h                   \
206         elf_386.h               \
207         elf_SPARC.h             \
208         elf_notes.h             \
209         elf_amd64.h             \
210         elftypes.h              \
211         emul64.h                \
212         emul64cmd.h             \
213         emul64var.h             \
214         epm.h                   \
215         errno.h                 \
216         errorq.h                \
217         errorq_impl.h           \
218         esunddi.h               \
219         ethernet.h              \
220         euc.h                   \
221         eucioctl.h              \
222         exacct.h                \
223         exacct_catalog.h        \
224         exacct_impl.h           \
225         exec.h                  \
226         exechdr.h               \
227         extdirent.h             \
228         fault.h                 \
229         fasttrap.h              \
230         fasttrap_impl.h         \
231         fbio.h                  \
232         fbuf.h                  \
233         fcntl.h                 \
234         fct.h                   \
235         fct_defines.h           \
236         fctio.h                 \
237         fdbuffer.h              \
238         fdio.h                  \
239         feature_tests.h         \
240         fem.h                   \
241         file.h                  \
242         filio.h                 \
243         flock.h                 \
244         flock_impl.h            \
245         fork.h                  \
246         fss.h                   \
247         fsspriocntl.h           \
248         fsid.h                  \
249         fssnap.h                \
250         fssnap_if.h             \
251         fstyp.h                 \
252         ftrace.h                \
253         fx.h                    \
254         fxpriocntl.h            \
255         gfs.h                   \
256         gld.h                   \
257         gldpriv.h               \
```

```
258         group.h                 \
259         hdio.h                  \
260         hook.h                  \
261         hook_event.h            \
262         hook_impl.h             \
263         hwconf.h                \
264         ia.h                    \
265         iapriocntl.h            \
266         ibpart.h                \
267         id32.h                  \
268         idmap.h                 \
269         ieeefp.h                \
270         id_space.h              \
271         instance.h              \
272         int_const.h             \
273         int_fmtio.h             \
274         int_limits.h            \
275         int_types.h             \
276         inttypes.h              \
277         ioccom.h                \
278         ioctl.h                 \
279         ipc.h                   \
280         ipc_impl.h              \
281         ipc_rctl.h              \
282         ipd.h                   \
283         ipmi.h                  \
284         isa_defs.h              \
285         iscsi_authclient.h      \
286         iscsi_authclientglue.h  \
287         iscsi_protocol.h        \
288         jioctl.h                \
289         kbd.h                   \
290         kbdreg.h                \
291         kbio.h                  \
292         kcpc.h                  \
293         kdi.h                   \
294         kdi_impl.h              \
295         kiconv.h                \
296         kiconv_big5_utf8.h      \
297         kiconv_cck_common.h     \
298         kiconv_cp950hkscs_utf8.h        \
299         kiconv_emea1.h          \
300         kiconv_emea2.h          \
301         kiconv_euckr_utf8.h     \
302         kiconv_euctw_utf8.h     \
303         kiconv_gb18030_utf8.h   \
304         kiconv_gb2312_utf8.h    \
305         kiconv_hkscs_utf8.h     \
306         kiconv_ja.h             \
307         kiconv_ja_jis_to_unicode.h      \
308         kiconv_ja_unicode_to_jis.h      \
309         kiconv_ko.h             \
310         kiconv_latin1.h         \
311         kiconv_sc.h             \
312         kiconv_tc.h             \
313         kiconv_uhc_utf8.h       \
314         kiconv_utf8_big5.h      \
315         kiconv_utf8_cp950hkscs.h        \
316         kiconv_utf8_euckr.h     \
317         kiconv_utf8_euctw.h     \
318         kiconv_utf8_gb18030.h   \
319         kiconv_utf8_gb2312.h    \
320         kiconv_utf8_hkscs.h     \
321         kiconv_utf8_uhc.h       \
322         kidmap.h                \
323         klpd.h                  \
```



```
324         klwp.h                  \
325         kmdb.h                  \
326         kmem.h                  \
327         kmem_impl.h             \
328         kobj.h                  \
329         kobj_impl.h             \
330         ksocket.h               \
331         kstat.h                 \
332         kstr.h                  \
333         ksyms.h                 \
334         ksynch.h                \
335         ldterm.h                \
336         lgrp.h                  \
337         lgrp_user.h             \
338         libc_kernel.h           \
339         link.h                  \
340         list.h                  \
341         list_impl.h             \
342         llc1.h                  \
343         loadavg.h               \
344         lock.h                  \
345         lockfs.h                \
346         lockstat.h              \
347         lofi.h                  \
348         log.h                   \
349         logindmux.h             \
350         logindmux_impl.h        \
351         lwp.h                   \
352         lwp_timer_impl.h        \
353         lwp_upimutex_impl.h     \
354         lpif.h                  \
355         mac.h                   \
356         mac_client.h            \
357         mac_client_impl.h       \
358         mac_ether.h             \
359         mac_flow.h              \
360         mac_flow_impl.h         \
361         mac_impl.h              \
362         mac_provider.h          \
363         mac_soft_ring.h         \
364         mac_stat.h              \
365         machelf.h               \
366         map.h                   \
367         md4.h                   \
368         md5.h                   \
369         md5_consts.h            \
370         mdi_impldefs.h          \
371         mem.h                   \
372         mem_config.h            \
373         memlist.h               \
374         mkdev.h                 \
375         mhd.h                   \
376         mii.h                   \
377         miiregs.h               \
378         mixer.h                 \
379         mman.h                  \
380         mmapobj.h               \
381         mntent.h                \
382         mntio.h                 \
383         mnttab.h                \
384         modctl.h                \
385         mode.h                  \
386         model.h                 \
387         modhash.h               \
388         modhash_impl.h          \
389         mount.h                 \
```

```
390         mouse.h                 \
391         msacct.h                \
392         msg.h                   \
393         msg_impl.h              \
394         msio.h                  \
395         msreg.h                 \
396         mtio.h                  \
397         multidata.h             \
398         multidata_impl.h        \
399         mutex.h                 \
400         nbmlock.h               \
401         ndifm.h                 \
402         ndi_impldefs.h          \
403         net80211.h              \
404         net80211_crypto.h       \
405         net80211_ht.h           \
406         net80211_proto.h        \
407         netconfig.h             \
408         neti.h                  \
409         netstack.h              \
410         nexusdefs.h             \
411         note.h                  \
412         null.h                  \
413         nvpair.h                \
414         nvpair_impl.h           \
415         objfs.h                 \
416         objfs_impl.h            \
417         ontrap.h                \
418         open.h                  \
419         openpromio.h            \
420         panic.h                 \
421         param.h                 \
422         pathconf.h              \
423         pathname.h              \
424         pattr.h                 \
425         queue.h                 \
426         serializer.h            \
427         pbio.h                  \
428         pccard.h                \
429         pci.h                   \
430         pcie.h                  \
431         pci_impl.h              \
432         pci_tools.h             \
433         pcmcia.h                \
434         pctypes.h               \
435         pfmod.h                 \
436         pg.h                    \
437         pghw.h                  \
438         physmem.h               \
439         pkp_hash.h              \
440         pm.h                    \
441         policy.h                \
442         poll.h                  \
443         poll_impl.h             \
444         pool.h                  \
445         pool_impl.h             \
446         pool_pset.h             \
447         port.h                  \
448         port_impl.h             \
449         port_kernel.h           \
450         portif.h                \
451         ppmio.h                 \
452         pppt_ic_if.h            \
453         pppt_ioctl.h            \
454         priocntl.h              \
455         priv.h                  \
```



```
456         priv_impl.h             \
457         prnio.h                 \
458         proc.h                  \
459         processor.h             \
460         procfs.h                \
461         procset.h               \
462         project.h               \
463         protosw.h               \
464         prsystm.h               \
465         pset.h                  \
466         pshot.h                 \
467         ptem.h                  \
468         ptms.h                  \
469         ptyvar.h                \
470         raidioctl.h             \
471         ramdisk.h               \
472         random.h                \
473         rctl.h                  \
474         rctl_impl.h             \
475         rds.h                   \
476         reboot.h                \
477         refstr.h                \
478         refstr_impl.h           \
479         resource.h              \
480         rlioctl.h               \
481         rt.h                    \
482         rtpriocntl.h            \
483         rwlock.h                \
484         rwlock_impl.h           \
485         rwstlock.h              \
486         sad.h                   \
487         schedctl.h              \
488         sdt.h                   \
489         select.h                \
490         sem.h                   \
491         sem_impl.h              \
492         sema_impl.h             \
493         semaphore.h             \
494         sendfile.h              \
495         ser_sync.h              \
496         session.h               \
497         sha1.h                  \
498         sha1_consts.h           \
499         sha2.h                  \
500         sha2_consts.h           \
501         share.h                 \
502         shm.h                   \
503         shm_impl.h              \
504         sid.h                   \
505         siginfo.h               \
506         signal.h                \
507         sleepq.h                \
508         smbios.h                \
509         smbios_impl.h           \
510         sobject.h               \
511         socket.h                \
512         socket_impl.h           \
513         socket_proto.h          \
514         socketvar.h             \
515         sockfilter.h            \
516         sockio.h                \
517         soundcard.h             \
518         squeue.h                \
519         squeue_impl.h           \
520         srn.h                   \
521         sservice.h              \
```

```
522         stat.h                  \
523         statfs.h                \
524         statvfs.h               \
525         stdbool.h               \
526         stdint.h                \
527         stermio.h               \
528         stmf.h                  \
529         stmf_defines.h          \
530         stmf_ioctl.h            \
531         stmf_sbd_ioctl.h        \
532         stream.h                \
533         strft.h                 \
534         strlog.h                \
535         strmdep.h               \
536         stropts.h               \
537         strredir.h              \
538         strstat.h               \
539         strsubr.h               \
540         strsun.h                \
541         strtty.h                \
542         sunddi.h                \
543         sunldi.h                \
544         sunldi_impl.h           \
545         sunmdi.h                \
546         sunndi.h                \
547         sunos_dhcp_class.h      \
548         sunpm.h                 \
549         suntpi.h                \
550         suntty.h                \
551         swap.h                  \
552         synch.h                 \
553         sysdc.h                 \
554         sysdc_impl.h            \
555         syscall.h               \
556         sysconf.h               \
557         sysconfig.h             \
558         sysevent.h              \
559         sysevent_impl.h         \
560         sysinfo.h               \
561         syslog.h                \
562         sysmacros.h             \
563         sysmsg_impl.h           \
564         systeminfo.h            \
565         systm.h                 \
566         task.h                  \
567         taskq.h                 \
568         taskq_impl.h            \
569         t_kuser.h               \
570         t_lock.h                \
571         telioctl.h              \
572         termio.h                \
573         termios.h               \
574         termiox.h               \
575         thread.h                \
576         ticlts.h                \
577         ticots.h                \
578         ticotsord.h             \
579         tihdr.h                 \
580         time.h                  \
581         time_impl.h             \
582         time_std_impl.h         \
583         timeb.h                 \
584         timer.h                 \
585         times.h                 \
586         timex.h                 \
587         timod.h                 \
```



```
588         tirdwr.h                \
589         tiuser.h                \
590         tl.h                    \
591         tnf.h                   \
592         tnf_com.h               \
593         tnf_probe.h             \
594         tnf_writer.h            \
595         todio.h                 \
596         tpicommon.h             \
597         ts.h                    \
598         tspriocntl.h            \
599         ttcompat.h              \
600         ttold.h                 \
601         tty.h                   \
602         ttychars.h              \
603         ttydev.h                \
604         tuneable.h              \
605         turnstile.h             \
606         types.h                 \
607         types32.h               \
608         tzfile.h                \
609         u8_textprep.h           \
610         u8_textprep_data.h      \
611         uadmin.h                \
612         ucred.h                 \
613         uio.h                   \
614         ulimit.h                \
615         un.h                    \
616         unistd.h                \
617         user.h                  \
618         ustat.h                 \
619         utime.h                 \
620         utsname.h               \
621         utssys.h                \
622         uuid.h                  \
623         va_impl.h               \
624         va_list.h               \
625         var.h                   \
626         varargs.h               \
627         vfs.h                   \
628         vfs_opreg.h             \
629         vfstab.h                \
630         vgareg.h                \
631         videodev2.h             \
632         visual_io.h             \
633         vlan.h                  \
634         vm.h                    \
635         vm_usage.h              \
636         vmem.h                  \
637         vmem_impl.h             \
638         vmsystm.h               \
639         vnic.h                  \
640         vnic_impl.h             \
641         vnode.h                 \
642         vscan.h                 \
643         vtoc.h                  \
644         vtrace.h                \
645         vuid_event.h            \
646         vuid_wheel.h            \
647         vuid_queue.h            \
648         vuid_state.h            \
649         vuid_store.h            \
650         wait.h                  \
651         waitq.h                 \
652         wanboot_impl.h          \
653         watchpoint.h            \
```

```
654         winlockio.h             \
655         zcons.h                 \
656         zone.h                  \
657         xti_inet.h              \
658         xti_osi.h               \
659         xti_xtiopt.h            \
660         zmod.h

662 HDRS=                           \
663         $(GENHDRS)              \
664         $(CHKHDRS)

666 AUDIOHDRS=                      \
667         ac97.h                  \
668         audio_common.h          \
669         audio_driver.h          \
670         audio_oss.h             \
671         g711.h

673 AVHDRS=                        \
674         iec61883.h

676 BSCHDRS=                       \
677         bscbus.h                \
678         bscv_impl.h             \
679         lom_ebuscodes.h         \
680         lom_io.h                \
681         lom_priv.h              \
682         lombus.h

684 MDESCHDRS=                     \
685         mdesc.h                 \
686         mdesc_impl.h

688 CPUDRVHDRS=                    \
689         cpudrv.h

691 CRYPTOHDRS=                    \
692         elfsign.h               \
693         ioctl.h                 \
694         ioctladmin.h            \
695         common.h                \
696         impl.h                  \
697         spi.h                   \
698         api.h                   \
699         ops_impl.h              \
700         sched_impl.h

702 DCAMHDRS=                      \
703         dcam1394_io.h

705 IBHDRS=                \
706         ib_types.h      \
707         ib_pkt_hdrs.h

709 IBTLHDRS=               \
710         ibtl_types.h    \
711         ibtl_status.h   \
712         ibti.h          \
713         ibti_cm.h       \
714         ibci.h          \
715         ibti_common.h   \
716         ibvti.h         \
717         ibtl_ci_types.h

719 IBTLIMPLHDRS=           \
```



```
720         ibtl_util.h

722 IBNEXHDRS=              \
723         ibnex_devctl.h

725 IBMFHDRS=               \
726         ibmf.h          \
727         ibmf_msg.h      \
728         ibmf_saa.h      \
729         ibmf_utils.h

731 IBMGTHDRS=              \
732         ib_dm_attr.h    \
733         ib_mad.h        \
734         sm_attr.h       \
735         sa_recs.h

737 IBDHDRS=                \
738         ibd.h

740 OFHDRS=                 \
741         ofa_solaris.h   \
742         ofed_kernel.h

744 RDMAHDRS=               \
745         ib_addr.h       \
746         ib_user_mad.h   \
747         ib_user_sa.h    \
748         ib_user_verbs.h \
749         ib_verbs.h      \
750         rdma_cm.h       \
751         rdma_user_cm.h

753 SOL_UVERBSHDRS=         \
754         sol_uverbs.h    \
755         sol_uverbs2ucma.h       \
756         sol_uverbs_comp.h       \
757         sol_uverbs_hca.h        \
758         sol_uverbs_qp.h         \
759         sol_uverbs_event.h

761 SOL_UMADHDRS=           \
762         sol_umad.h

764 SOL_UCMAHDRS=           \
765         sol_ucma.h      \
766         sol_rdma_user_cm.h

768 SOL_OFSHDRS=            \
769         sol_cma.h       \
770         sol_ib_cma.h    \
771         sol_ofs_common.h        \
772         sol_kverb_impl.h

774 TAVORHDRS=              \
775         tavor_ioctl.h

777 HERMONHDRS=             \
778         hermon_ioctl.h

780 MLNXHDRS=              \
781         mlnx_umap.h

783 IDMHDRS=                \
784         idm.h           \
785         idm_impl.h      \
```

```
786         idm_so.h        \
787         idm_text.h      \
788         idm_transport.h \
789         idm_conn_sm.h

791 ISCSITHDRS=             \
792         radius_packet.h         \
793         radius_protocol.h       \
794         chap.h                  \
795         isns_protocol.h         \
796         iscsi_if.h              \
797         iscsit_common.h

799 ISOHDRS=                        \
800         signal_iso.h

802 DERIVED_LVMHDRS=        \
803         md_mdiox.h              \
804         md_basic.h              \
805         mdmed.h                 \
806         md_mhdx.h               \
807         mdmn_commd.h

809 LVMHDRS=                        \
810         md_convert.h            \
811         md_crc.h                \
812         md_hotspares.h          \
813         md_mddb.h               \
814         md_mirror.h             \
815         md_mirror_shared.h      \
816         md_names.h              \
817         md_notify.h             \
818         md_raid.h               \
819         md_rename.h             \
820         md_sp.h                 \
821         md_stripe.h             \
822         md_trans.h              \
823         mdio.h                  \
824         mdvar.h

826 ALL_LVMHDRS=                    \
827         $(LVMHDRS)              \
828         $(DERIVED_LVMHDRS)

830 FMHDRS=                         \
831         protocol.h              \
832         util.h

834 FMFSHDRS=                       \
835         zfs.h

837 FMIOHDRS=                       \
838         ddi.h                   \
839         disk.h                  \
840         pci.h                   \
841         scsi.h                  \
842         sun4upci.h              \
843         opl_mc_fm.h

845 FSHDRS=                         \
846         autofs.h                \
847         cachefs_dir.h           \
848         cachefs_dlog.h          \
849         cachefs_filegrp.h       \
850         cachefs_fs.h            \
851         cachefs_fscache.h       \
```



```
852         cachefs_ioctl.h         \
853         cachefs_log.h           \
854         decomp.h                \
855         dv_node.h               \
856         sdev_impl.h             \
857         fifonode.h              \
858         hsfs_isospec.h          \
859         hsfs_node.h             \
860         hsfs_rrip.h             \
861         hsfs_spec.h             \
862         hsfs_susp.h             \
863         lofs_info.h             \
864         lofs_node.h             \
865         mntdata.h               \
866         namenode.h              \
867         pc_dir.h                \
868         pc_fs.h                 \
869         pc_label.h              \
870         pc_node.h               \
871         pxfs_ki.h               \
872         snode.h                 \
873         swapnode.h              \
874         tmp.h                   \
875         tmpnode.h               \
876         udf_inode.h             \
877         udf_volume.h            \
878         ufs_acl.h               \
879         ufs_bio.h               \
880         ufs_filio.h             \
881         ufs_fs.h                \
882         ufs_fsdir.h             \
883         ufs_inode.h             \
884         ufs_lockfs.h            \
885         ufs_log.h               \
886         ufs_mount.h             \
887         ufs_panic.h             \
888         ufs_prot.h              \
889         ufs_quota.h             \
890         ufs_snap.h              \
891         ufs_trans.h             \
892         zfs.h                   \
893         zut.h

895 SCSIHDRS=              \
896         scsi.h          \
897         scsi_address.h  \
898         scsi_ctl.h      \
899         scsi_fm.h       \
900         scsi_params.h   \
901         scsi_pkt.h      \
902         scsi_resource.h \
903         scsi_types.h    \
904         scsi_watch.h

906 SCSICONFHDRS=          \
907         autoconf.h      \
908         device.h

910 SCSIGENHDRS=           \
911         commands.h      \
912         dad_mode.h      \
913         inquiry.h       \
914         message.h       \
915         mode.h          \
916         persist.h       \
917         sense.h         \
```

```
 918         sff_frames.h    \
 919         smp_frames.h    \
 920         status.h

 922 SCSIIMPLHDRS=           \
 923         commands.h      \
 924         inquiry.h       \
 925         mode.h          \
 926         scsi_reset_notify.h \
 927         scsi_sas.h      \
 928         sense.h         \
 929         services.h      \
 930         smp_transport.h \
 931         spc3_types.h    \
 932         status.h        \
 933         transport.h     \
 934         types.h         \
 935         uscsi.h         \
 936         usmp.h

 938 SCSITARGETSHDRS=        \
 939         ses.h           \
 940         sesio.h         \
 941         sgendef.h       \
 942         stdef.h         \
 943         sddef.h         \
 944         smp.h

 946 SCSIADHDRS=

 948 SCSICADHDRS=

 950 SCSIISCSIHDRS=          \
 951         iscsi_door.h    \
 952         iscsi_if.h

 954 SCSIVHCIHDRS=           \
 955         scsi_vhci.h     \
 956         mpapi_impl.h    \
 957         mpapi_scsi_vhci.h

 959 SDCARDHDRS=            \
 960         sda.h           \
 961         sda_impl.h      \
 962         sda_ioctl.h

 964 FC4HDRS=                       \
 965         fc_transport.h  \
 966         linkapp.h       \
 967         fc.h            \
 968         fcp.h           \
 969         fcal_transport.h \
 970         fcal.h          \
 971         fcal_linkapp.h  \
 972         fcio.h

 974 FCHDRS=                 \
 975         fc.h            \
 976         fcio.h          \
 977         fc_types.h      \
 978         fc_appif.h

 980 FCIMPLHDRS=            \
 981         fc_error.h      \
 982         fcph.h
```



```
 984 FCULPHDRS=             \
 985         fcp_util.h      \
 986         fcsm.h

 988 SATAGENHDRS=           \
 989         sata_hba.h      \
 990         sata_defs.h     \
 991         sata_cfgadm.h

 993 SYSEVENTHDRS=          \
 994         ap_driver.h     \
 995         dev.h           \
 996         domain.h        \
 997         dr.h            \
 998         env.h           \
 999         eventdefs.h     \
1000         ipmp.h          \
1001         pwrctl.h        \
1002         svm.h           \
1003         vrrp.h

1005 CONTRACTHDRS=           \
1006         process.h       \
1007         process_impl.h  \
1008         device.h        \
1009         device_impl.h

1011 USBHDRS=        \
1012         usba.h  \
1013         usbai.h

1015 UWBHDRS=        \
1016         uwb.h   \
1017         uwbai.h

1019 UWBAHDRS=       \
1020         uwba.h

1022 USBAUDHDRS=           \
1023         usb_audio.h

1025 USBHUBDHDRS=          \
1026         hub.h          \
1027         hubd_impl.h

1029 USBHIDHDRS=           \
1030         hid.h

1032 USBHWARCHDRS=         \
1033         hwarc.h

1035 USBMSHDRS=            \
1036         usb_bulkonly.h  \
1037         usb_cbi.h

1039 USBPRNHDRS=          \
1040         usb_printer.h

1042 USBCDCHDRS=           \
1043         usb_cdc.h

1045 USBVIDHDRS=          \
1046         usbvc.h

1048 USBWCMHDRS=           \
1049         usbwcm.h
```

```
1051 UGENHDRS=           \
1052         usb_ugen.h

1054 HOTPLUGHDRS=    \
1055         hpcsvc.h        \
1056         hpctrl.h

1058 HOTPLUGPCIHDRS= \
1059         pcicfg.h        \
1060         pcihp.h

1062 RSMHDRS= \
1063         rsm.h   \
1064         rsm_common.h    \
1065         rsmapi_common.h \
1066         rsmpi.h \
1067         rsmpi_driver.h  \
1068         rsmka_path_int.h

1070 TSOLHDRS=               \
1071         label.h         \
1072         label_macro.h   \
1073         priv.h          \
1074         tndb.h          \
1075         tsyscall.h

1077 I1394HDRS=              \
1078         cmd1394.h       \
1079         id1394.h        \
1080         ieee1212.h      \
1081         ieee1394.h      \
1082         ixl1394.h       \
1083         s1394_impl.h    \
1084         t1394.h

1086 # "cmdk" headers used on sparc
1087 SDKTPHDRS=              \
1088         dadkio.h        \
1089         fdisk.h

1091 # "cmdk" headers used on i386
1092 DKTPHDRS=               \
1093         altsctr.h       \
1094         bbh.h           \
1095         cm.h            \
1096         cmdev.h         \
1097         cmdk.h          \
1098         cmpkt.h         \
1099         controller.h    \
1100         dadev.h         \
1101         dadk.h          \
1102         dadkio.h        \
1103         fctypes.h       \
1104         fdisk.h         \
1105         flowctrl.h      \
1106         gda.h           \
1107         quetypes.h      \
1108         queue.h         \
1109         tgcom.h         \
1110         tgdk.h

1112 # "pc" header files used on i386
1113 PCHDRS=                 \
1114         avintr.h        \
1115         dma_engine.h    \
```



```
1116         i8272A.h        \
1117         pcic_reg.h      \
1118         pcic_var.h      \
1119         pic.h           \
1120         pit.h           \
1121         rtc.h

1123 NXGEHDRS=                       \
1124         nxge.h                  \
1125         nxge_common.h           \
1126         nxge_common_impl.h      \
1127         nxge_defs.h             \
1128         nxge_hw.h               \
1129         nxge_impl.h             \
1130         nxge_ipp.h              \
1131         nxge_ipp_hw.h           \
1132         nxge_mac.h              \
1133         nxge_mac_hw.h           \
1134         nxge_fflp.h             \
1135         nxge_fflp_hw.h          \
1136         nxge_mii.h              \
1137         nxge_rxdma.h            \
1138         nxge_rxdma_hw.h         \
1139         nxge_txc.h              \
1140         nxge_txc_hw.h           \
1141         nxge_txdma.h            \
1142         nxge_txdma_hw.h         \
1143         nxge_virtual.h          \
1144         nxge_espc.h

1146 include Makefile.syshdrs

1148 dcam/%.check:   dcam/%.h
1149         $(DOT_H_CHECK)

1151 CHECKHDRS=                                                      \
1152         $($(MACH)_HDRS:%.h=%.check)                             \
1153         $(AUDIOHDRS:%.h=audio/%.check)                          \
1154         $(AVHDRS:%.h=av/%.check)                                \
1155         $(BSCHDRS:%.h=%.check)                                  \
1156         $(CHKHDRS:%.h=%.check)                                  \
1157         $(CPUDRVHDRS:%.h=%.check)                               \
1158         $(CRYPTOHDRS:%.h=crypto/%.check)                        \
1159         $(DCAMHDRS:%.h=dcam/%.check)                            \
1160         $(FC4HDRS:%.h=fc4/%.check)                              \
1161         $(FCHDRS:%.h=fibre-channel/%.check)                     \
1162         $(FCIMPLHDRS:%.h=fibre-channel/impl/%.check)            \
1163         $(FCULPHDRS:%.h=fibre-channel/ulp/%.check)              \
1164         $(IBHDRS:%.h=ib/%.check)                                \
1165         $(IBDHDRS:%.h=ib/clients/ibd/%.check)                   \
1166         $(IBTLHDRS:%.h=ib/ibtl/%.check)                         \
1167         $(IBTLIMPLHDRS:%.h=ib/ibtl/impl/%.check)                \
1168         $(IBNEXHDRS:%.h=ib/ibnex/%.check)               \
1169         $(IBMGTHDRS:%.h=ib/mgt/%.check)                         \
1170         $(IBMFHDRS:%.h=ib/mgt/ibmf/%.check)                     \
1171         $(OFHDRS:%.h=ib/clients/of/%.check)                             \
1172         $(RDMAHDRS:%.h=ib/clients/of/rdma/%.check)                      \
1173         $(SOL_UVERBSHDRS:%.h=ib/clients/of/sol_uverbs/%.check)  \
1174         $(SOL_UCMAHDRS:%.h=ib/clients/of/sol_ucma/%.check)      \
1175         $(SOL_OFSHDRS:%.h=ib/clients/of/sol_ofs/%.check)        \
1176         $(TAVORHDRS:%.h=ib/adapters/tavor/%.check)              \
1177         $(HERMONHDRS:%.h=ib/adapters/hermon/%.check)            \
1178         $(MLNXHDRS:%.h=ib/adapters/%.check)                     \
1179         $(IDMHDRS:%.h=idm/%.check)                              \
1180         $(ISCSIHDRS:%.h=iscsi/%.check)                          \
1181         $(ISCSITHDRS:%.h=iscsit/%.check)                                \
```

```
1182         $(ISOHDRS:%.h=iso/%.check)                     \
1183         $(FMHDRS:%.h=fm/%.check)                       \
1184         $(FMFSHDRS:%.h=fm/fs/%.check)                  \
1185         $(FMIOHDRS:%.h=fm/io/%.check)                  \
1186         $(FSHDRS:%.h=fs/%.check)                       \
1187         $(LVMHDRS:%.h=lvm/%.check)                     \
1188         $(SCSIHDRS:%.h=scsi/%.check)                   \
1189         $(SCSIADHDRS:%.h=scsi/adapters/%.check)        \
1190         $(SCSICONFHDRS:%.h=scsi/conf/%.check)          \
1191         $(SCSIIMPLHDRS:%.h=scsi/impl/%.check)          \
1192         $(SCSIISCSIHDRS:%.h=scsi/adapters/%.check)     \
1193         $(SCSIGENHDRS:%.h=scsi/generic/%.check)        \
1194         $(SCSITARGETSHDRS:%.h=scsi/targets/%.check)    \
1195         $(SCSIVHCIHDRS:%.h=scsi/adapters/%.check)      \
1196         $(SATAGENHDRS:%.h=sata/%.check)                \
1197         $(SDCARDHDRS:%.h=sdcard/%.check)               \
1198         $(SYSEVENTHDRS:%.h=sysevent/%.check)           \
1199         $(CONTRACTHDRS:%.h=contract/%.check)           \
1200         $(USBAUDHDRS:%.h=usb/clients/audio/%.check)   \
1201         $(USBHUBDHDRS:%.h=usb/hubd/%.check)            \
1202         $(USBHIDHDRS:%.h=usb/clients/hid/%.check)      \
1203         $(USBHWARCHDRS:%.h=usb/clients/hwarc/%.check)   \
1204         $(USBMSHDRS:%.h=usb/clients/mass_storage/%.check)       \
1205         $(USBPRNHDRS:%.h=usb/clients/printer/%.check)   \
1206         $(USBCDCHDRS:%.h=usb/clients/usbcdc/%.check)    \
1207         $(USBVIDHDRS:%.h=usb/clients/video/usbvc/%.check)   \
1208         $(USBWCMHDRS:%.h=usb/clients/usbinput/usbwcm/%.check)   \
1209         $(UGENHDRS:%.h=usb/clients/ugen/%.check)   \
1210         $(USBHDRS:%.h=usb/%.check)                     \
1211         $(UWBHDRS:%.h=uwb/%.check)                     \
1212         $(UWBAHDRS:%.h=uwb/uwba/%.check)                    \
1213         $(I1394HDRS:%.h=1394/%.check)                  \
1214         $(RSMHDRS:%.h=rsm/%.check)                     \
1215         $(TSOLHDRS:%.h=tsol/%.check)                   \
1216         $(NXGEHDRS:%.h=nxge/%.check)


1219 .KEEP_STATE:

1221 .PARALLEL:                      \
1222         $(CHECKHDRS)            \
1223         $(ROOTHDRS)             \
1224         $(ROOTAUDHDRS)          \
1225         $(ROOTAVHDRS)           \
1226         $(ROOTCRYPTOHDRS)       \
1227         $(ROOTDCAMHDRS)         \
1228         $(ROOTISOHDRS)          \
1229         $(ROOTIDMHDRS)          \
1230         $(ROOTISCSIHDRS)        \
1231         $(ROOTISCSITHDRS)       \
1232         $(ROOTFC4HDRS)          \
1233         $(ROOTFCHDRS)           \
1234         $(ROOTFCIMPLHDRS)       \
1235         $(ROOTFCULPHDRS)        \
1236         $(ROOTFMHDRS)           \
1237         $(ROOTFMIOHDRS)         \
1238         $(ROOTFMFSHDRS)         \
1239         $(ROOTFSHDRS)           \
1240         $(ROOTIBDHDRS)          \
1241         $(ROOTIBHDRS)           \
1242         $(ROOTIBTLHDRS)         \
1243         $(ROOTIBTLIMPLHDRS)     \
1244         $(ROOTIBNEXHDRS)        \
1245         $(ROOTIBMGTHDRS)        \
1246         $(ROOTIBMFHDRS)         \
1247         $(ROOTOFHDRS)           \
```



```
1248         $(ROOTRDMAHDRS)         \
1249         $(ROOTSOL_OFSHDRS)      \
1250         $(ROOTSOL_UMADHDRS)     \
1251         $(ROOTSOL_UVERBSHDRS)   \
1252         $(ROOTSOL_UCMAHDRS)     \
1253         $(ROOTTAVORHDRS)        \
1254         $(ROOTHERMONHDRS)       \
1255         $(ROOTMLNXHDRS)         \
1256         $(ROOTLVMHDRS)          \
1257         $(ROOTSCSIHDRS)         \
1258         $(ROOTSCSIADHDRS)       \
1259         $(ROOTSCSICONFHDRS)     \
1260         $(ROOTSCSIISCSIHDRS)    \
1261         $(ROOTSCSIGENHDRS)      \
1262         $(ROOTSCSIIMPLHDRS)     \
1263         $(ROOTSCSIVHCIHDRS)     \
1264         $(ROOTSDCARDHDRS)       \
1265         $(ROOTSYSEVENTHDRS)     \
1266         $(ROOTCONTRACTHDRS)     \
1267         $(ROOTUSBHDRS)          \
1268         $(ROOTUWBHDRS)          \
1269         $(ROOTUWBAHDRS)         \
1270         $(ROOTUSBAUDHDRS)       \
1271         $(ROOTUSBHUBDHDRS)      \
1272         $(ROOTUSBHIDHDRS)       \
1273         $(ROOTUSBHRCHDRS)       \
1274         $(ROOTUSBMSHDRS)        \
1275         $(ROOTUSBPRNHDRS)       \
1276         $(ROOTUSBCDCHDRS)       \
1277         $(ROOTUSBVIDHDRS)       \
1278         $(ROOTUSBWCMHDRS)       \
1279         $(ROOTUGENHDRS)         \
1280         $(ROOT1394HDRS)         \
1281         $(ROOTHOTPLUGHDRS)      \
1282         $(ROOTHOTPLUGPCIHDRS)   \
1283         $(ROOTRSMHDRS)          \
1284         $(ROOTTSOLHDRS)         \
1285         $($(MACH)_ROOTHDRS)


1288 install_h:                      \
1289         $(ROOTDIRS)             \
1290         LVMDERIVED_H            \
1291         .WAIT                   \
1292         $(ROOTHDRS)             \
1293         $(ROOTAUDHDRS)          \
1294         $(ROOTAVHDRS)           \
1295         $(ROOTCRYPTOHDRS)       \
1296         $(ROOTDCAMHDRS)         \
1297         $(ROOTISOHDRS)          \
1298         $(ROOTIDMHDRS)          \
1299         $(ROOTISCSIHDRS)        \
1300         $(ROOTISCSITHDRS)       \
1301         $(ROOTFC4HDRS)          \
1302         $(ROOTFCHDRS)           \
1303         $(ROOTFCIMPLHDRS)       \
1304         $(ROOTFCULPHDRS)        \
1305         $(ROOTFMHDRS)           \
1306         $(ROOTFMFSHDRS)         \
1307         $(ROOTFMIOHDRS)         \
1308         $(ROOTFSHDRS)           \
1309         $(ROOTIBDHDRS)          \
1310         $(ROOTIBHDRS)           \
1311         $(ROOTIBTLHDRS)         \
1312         $(ROOTIBTLIMPLHDRS)     \
1313         $(ROOTIBNEXHDRS)        \
```

```
1314         $(ROOTIBMGTHDRS)        \
1315         $(ROOTIBMFHDRS)         \
1316         $(ROOTOFHDRS)           \
1317         $(ROOTRDMAHDRS)         \
1318         $(ROOTSOL_OFSHDRS)      \
1319         $(ROOTSOL_UMADHDRS)     \
1320         $(ROOTSOL_UVERBSHDRS)   \
1321         $(ROOTSOL_UCMAHDRS)     \
1322         $(ROOTTAVORHDRS)        \
1323         $(ROOTHERMONHDRS)       \
1324         $(ROOTMLNXHDRS)         \
1325         $(ROOTLVMHDRS)          \
1326         $(ROOTSCSIHDRS)         \
1327         $(ROOTSCSIADHDRS)       \
1328         $(ROOTSCSIISCSIHDRS)    \
1329         $(ROOTSCSICONFHDRS)     \
1330         $(ROOTSCSIGENHDRS)      \
1331         $(ROOTSCSIIMPLHDRS)     \
1332         $(ROOTSCSIVHCIHDRS)     \
1333         $(ROOTSDCARDHDRS)       \
1334         $(ROOTSYSEVENTHDRS)     \
1335         $(ROOTCONTRACTHDRS)     \
1336         $(ROOTUWBHDRS)          \
1337         $(ROOTUWBAHDRS)         \
1338         $(ROOTUSBHDRS)          \
1339         $(ROOTUSBAUDHDRS)       \
1340         $(ROOTUSBHUBDHDRS)      \
1341         $(ROOTUSBHIDHDRS)       \
1342         $(ROOTUSBHRCHDRS)       \
1343         $(ROOTUSBMSHDRS)        \
1344         $(ROOTUSBPRNHDRS)       \
1345         $(ROOTUSBCDCHDRS)       \
1346         $(ROOTUSBVIDHDRS)       \
1347         $(ROOTUSBWCMHDRS)       \
1348         $(ROOTUGENHDRS)         \
1349         $(ROOT1394HDRS)         \
1350         $(ROOTHOTPLUGHDRS)      \
1351         $(ROOTHOTPLUGPCIHDRS)   \
1352         $(ROOTRSMHDRS)          \
1353         $(ROOTTSOLHDRS)         \
1354         $($(MACH)_ROOTHDRS)

1356 all_h: $(GENHDRS)

1358 priv_const.h: $(PRIVS_AWK) $(PRIVS_DEF)
1359         $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v privhfile=$@

1361 priv_names.h: $(PRIVS_AWK) $(PRIVS_DEF)
1362         $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v pubhfile=$@

1364 usb/usbdevs.h: $(USBDEVS_AWK) $(USBDEVS_DATA)
1365         $(NAWK) -f $(USBDEVS_AWK) $(USBDEVS_DATA) -H > $@

1367 LVMDERIVED_H:
1368         cd $(SRC)/uts/common/sys/lvm; pwd; $(MAKE) all_h

1370 clean:
1371         $(RM) $(GENHDRS)

1373 clobber: clean
1374         cd $(SRC)/uts/common/sys/lvm; pwd; $(MAKE) clobber

1376 check:  $(CHECKHDRS)

1378 FRC:
```

```
**********************************************************
    1346 Sun Dec 14 23:31:27 2014
new/usr/src/uts/common/sys/null.h
5218 posix definition of NULL
correct unistd.h and iso/stddef_iso.h
update gate source affected
**********************************************************
```

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

  22 /*
  23  * Copyright (c) 2003 Marcel Moolenaar.  All rights reserved.
  24  * Copyright 2014 PALO, Richard.
  25  */

  27 #ifndef _SYS_NULL_H
  28 #define _SYS_NULL_H

  30 #ifndef NULL

  32 #if !defined(__cplusplus)
  33 #define NULL    ((void *)0)
  34 #else
  35 #if __cplusplus >= 201103L
  36 #define NULL    nullptr
  37 #elif defined(__GNUG__) && defined(__GNUC__) && __GNUC__ >= 4
  38 #define NULL    __null
  39 #else
  40 #if defined(_LP64)
  41 #define NULL    (0L)
  42 #else
  43 #define NULL    0
  44 #endif  /* _LP64 */
  45 #endif  /* __GNUG__ */
  46 #endif  /* !__cplusplus */

  48 #endif

  50 #endif  /* _SYS_NULL_H */
```