```
**********************************************************
   32811 Wed Aug 29 09:52:21 2012
new/usr/src/cmd/svc/startd/method.c
3121 missing SMF method directories should say something useful
Reviewed by: Gary Mills <gary_mills@fastmail.fm>
Reviewed by: T Nguyen <truongqnguien@gmail.com>
Reviewed by: Richard Elling <richard.elling@gmail.com>
**********************************************************
______unchanged_portion_omitted_

 439 static void
 440 exec_method(const restarter_inst_t *inst, int type, const char *method,
 441     struct method_context *mcp, uint8_t need_session)
 442 {
 443         char *cmd;
 444         const char *errf;
 445         char **nenv;
 446         int rsmc_errno = 0;

 448         cmd = uu_msprintf("exec %s", method);

 450         if (inst->ri_utmpx_prefix[0] != '\0' && inst->ri_utmpx_prefix != NULL)
 451                 (void) utmpx_mark_init(getpid(), inst->ri_utmpx_prefix);

 453         setlog(inst->ri_logstem);
 454         log_instance(inst, B_FALSE, "Executing %s method (\"%s\").",
 455             method_names[type], method);

 457         if (need_session)
 458                 (void) setpgrp();

 460         /* Set credentials. */
 461         rsmc_errno = restarter_set_method_context(mcp, &errf);
 462         if (rsmc_errno != 0) {
 463                 log_instance(inst, B_FALSE,
 464                     "svc.startd could not set context for method: ");

 466                 if (rsmc_errno == -1) {
 467                         if (strcmp(errf, "core_set_process_path") == 0) {
 468                                 log_instance(inst, B_FALSE,
 469                                     "Could not set corefile path.");
 470                         } else if (strcmp(errf, "setproject") == 0) {
 471                                 log_instance(inst, B_FALSE, "%s: a resource "
 472                                     "control assignment failed", errf);
 473                         } else if (strcmp(errf, "pool_set_binding") == 0) {
 474                                 log_instance(inst, B_FALSE, "%s: a system "
 475                                     "error occurred", errf);
 476                         } else {
 477 #ifndef NDEBUG
 478                                 uu_warn("%s:%d: Bad function name \"%s\" for "
 479                                     "error %d from "
 480                                     "restarter_set_method_context().\n",
 481                                     __FILE__, __LINE__, errf, rsmc_errno);
 482 #endif
 483                                 abort();
 484                         }

 486                         exit(1);
 487                 }

 489                 if (errf != NULL && strcmp(errf, "pool_set_binding") == 0) {
 490                         switch (rsmc_errno) {
 491                         case ENOENT:
 492                                 log_instance(inst, B_FALSE, "%s: the pool "
 493                                     "could not be found", errf);
 494                                 break;
```



```
 496                         case EBADF:
 497                                 log_instance(inst, B_FALSE, "%s: the "
 498                                     "configuration is invalid", errf);
 499                                 break;

 501                         case EINVAL:
 502                                 log_instance(inst, B_FALSE, "%s: pool name "
 503                                     "\"%s\" is invalid", errf,
 504                                     mcp->resource_pool);
 505                                 break;

 507                         default:
 508 #ifndef NDEBUG
 509                                 uu_warn("%s:%d: Bad error %d for function %s "
 510                                     "in restarter_set_method_context().\n",
 511                                     __FILE__, __LINE__, rsmc_errno, errf);
 512 #endif
 513                                 abort();
 514                         }

 516                         exit(SMF_EXIT_ERR_CONFIG);
 517                 }

 519                 if (errf != NULL && strcmp(errf, "chdir") == 0) {
 520                         switch (rsmc_errno) {
 521                         case EACCES:
 522                         case EFAULT:
 523                         case EIO:
 524                         case ELOOP:
 525                         case ENAMETOOLONG:
 526                         case ENOENT:
 527                         case ENOLINK:
 528                         case ENOTDIR:
 529                                 log_instance(inst, B_FALSE, "%s: %s (\"%s\")",
 530                                     errf,
 531                                     strerror(rsmc_errno), mcp->working_dir);
 532                                 break;

 534                         default:
 535 #ifndef NDEBUG
 536                                 uu_warn("%s:%d: Bad error %d for function %s "
 537                                     "in restarter_set_method_context().\n",
 538                                     __FILE__, __LINE__, rsmc_errno, errf);
 539 #endif
 540                                 abort();
 541                         }

 543                         exit(SMF_EXIT_ERR_CONFIG);
 544                 }

 546                 if (errf != NULL) {
 547                         errno = rsmc_errno;
 548                         perror(errf);

 550                         switch (rsmc_errno) {
 551                         case EINVAL:
 552                         case EPERM:
 553                         case ENOENT:
 554                         case ENAMETOOLONG:
 555                         case ERANGE:
 556                         case ESRCH:
 557                                 exit(SMF_EXIT_ERR_CONFIG);
 558                                 /* NOTREACHED */

 560                         default:
```

```
561                                     exit(1);
562                             }
563                     }

565                     switch (rsmc_errno) {
566                     case ENOMEM:
567                             log_instance(inst, B_FALSE, "Out of memory.");
568                             exit(1);
569                             /* NOTREACHED */

571                     case ENOENT:
572                             log_instance(inst, B_FALSE, "Missing passwd entry for "
573                                 "user.");
574                             exit(SMF_EXIT_ERR_CONFIG);
575                             /* NOTREACHED */

577                     default:
578 #ifndef NDEBUG
579                             uu_warn("%s:%d: Bad miscellaneous error %d from "
580                                 "restarter_set_method_context().\n", __FILE__,
581                                 __LINE__, rsmc_errno);
582 #endif
583                             abort();
584                     }
585             }

587             nenv = set_smf_env(mcp->env, mcp->env_sz, NULL, inst,
588                 method_names[type]);

590             log_preexec();

592             (void) execle(SBIN_SH, SBIN_SH, "-c", cmd, NULL, nenv);

594             exit(10);
595 }
```

*unchanged_portion_omitted_*