

new/exception_lists/packaging

```
*****
27859 Mon Jul 17 16:15:15 2017
new/exception_lists/packaging
8330 Add svc_tp_create_addr to libnsl
Reviewed by: Paul Dagnelie <pcd@delphix.com>
Reviewed by: Evan Layton <evan.layton@nexenta.com>
Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>
*****
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 2012 OmniTI Computer Consulting, Inc. All rights reserved.
25 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
26 # Copyright 2016 Toomas Soome <tsoome@me.com>
27 # Copyright 2016 Hans Rosenfeld <rosenfeld@grumpf.hope-2000.org>
28 # Copyright 2016 Nexenta Systems, Inc.
29 #

31 #
32 # Exception List for validate_pkg
33 #

35 #
36 # The following entries are built in the /proto area
37 # but not included in any packages - this is intentional.
38 #
39 usr/include/auth_list.h
40 usr/include/bsm/audit_door_infc.h
41 usr/include/bsm/audit_private.h
42 usr/include/bsm/devalloc.h
43 usr/include/getxby_door.h
44 usr/include/passwdutil.h
45 usr/include/priv_utils.h
46 usr/include/rpcsvc/daemon_utils.h
47 usr/include/rpcsvc/svc_dg_priv.h
48 usr/include/security/pam_impl.h
49 usr/include/sys/clock_impl.h
50 usr/include/sys/winlockio.h
51 usr/include/scsi/plugins/ses/vendor/sun_impl.h
52 #
53 # Private lofi interface.
54 #
55 usr/include/sys/lofi_impl.h
56 #
57 # Private/Internal libraries of the Cryptographic Framework.
58 #
```

1

new/exception_lists/packaging

```
59 lib/libkcfid.so
60 lib/llib-lelfsign
61 lib/llib-lelfsign.ln
62 lib/llib-lkcfid
63 lib/llib-lkcfid.ln
64 usr/include/libelfsign.h
65 usr/lib/llib-lsoftcrypto
66 usr/lib/llib-lsoftcrypto.ln
67 usr/lib/amd64/llib-lsoftcrypto.ln      i386
68 usr/lib/sparcv9/llib-lsoftcrypto.ln    sparc

70 #
71 # The following files are used by the DHCP service, the
72 # standalone's DHCP implementation, and the kernel (nfs_dlboot).
73 # They contain interfaces which are currently private.
74 #
75 usr/include/dhcp_symbol.h
76 usr/include/sys/sunos_dhcp_class.h
77 #
78 # Private MAC driver header files
79 #
80 usr/include/inet/iptun.h
81 usr/include/sys/aggr_impl.h
82 usr/include/sys/aggr.h
83 usr/include/sys/dld_impl.h
84 usr/include/sys/dld_ioc.h
85 usr/include/sys/dls_impl.h
86 usr/include/sys/dls.h
87 usr/include/sys/mac_client_impl.h
88 usr/include/sys/mac_client.h
89 usr/include/sys/mac_flow_impl.h
90 usr/include/sys/mac_impl.h
91 usr/include/sys/mac_soft_ring.h
92 usr/include/sys/mac_stat.h
93 #
94 # Private GLDv3 userland libraries and headers
95 #
96 usr/include/libdladm_impl.h
97 usr/include/libdlaggr.h
98 usr/include/libdether.h
99 usr/include/libdlflow_impl.h
100 usr/include/libdlflow.h
101 usr/include/libdliptun.h
102 usr/include/libdlmgmt.h
103 usr/include/libdlsim.h
104 usr/include/libdlstat.h
105 usr/include/libdlvnic.h
106 usr/include/libdlwlan_impl.h
107 usr/include/libdlwlan.h
108 #
109 # Virtual Network Interface Card (VNIC)
110 #
111 usr/include/sys/vnic.h
112 usr/include/sys/vnic_impl.h
113 #
114 # Private libipadm lint library and header files
115 #
116 usr/include/ipadm_ipmgmt.h
117 usr/include/ipadm_ndpd.h
118 usr/include/libipadm.h
119 lib/llib-libipadm
120 lib/llib-libipadm.ln
121 lib/libipadm.so
122 #
123 # Private libsocket header file
124 #
```

2

new/exception_lists/packaging

```

125 usr/include/libsocket_priv.h
126 #
127 # IKE and IPsec support library exceptions. The IKE support
128 # library contains exclusively private interfaces, as does
129 # libipsecutil. My apologies for the glut of header files here.
130 #
131 usr/include/errfp.h
132 usr/include/ikedoor.h
133 usr/include/ipsec_util.h
134 usr/lib/libike.so
135 usr/lib/amd64/libike.so i386
136 usr/lib/sparcv9/libike.so sparc
137 usr/lib/libipsecutil.so
138 usr/lib/amd64/libipsecutil.so i386
139 usr/lib/sparcv9/libipsecutil.so sparc
140 usr/lib/llib-like
141 usr/lib/llib-like.ln
142 usr/lib/amd64/llib-like.ln i386
143 usr/lib/sparcv9/llib-like.ln sparc
144 usr/lib/llib-lipsecutil
145 usr/lib/llib-lipsecutil.ln
146 usr/lib/amd64/llib-lipsecutil.ln i386
147 usr/lib/sparcv9/llib-lipsecutil.ln sparc
148 #
149 usr/include/inet/ip_impl.h
150 usr/include/inet/ip_ndp.h
151 usr/include/inet/ip2mac_impl.h
152 usr/include/inet/ip2mac.h
153 usr/include/inet/rawip_impl.h
154 usr/include/inet/tcp_impl.h
155 usr/include/inet/udp_impl.h
156 usr/include/libmail.h
157 usr/include/libwam_priv.h
158 usr/include/protocols/ripngd.h
159 usr/include/s_string.h
160 usr/include/sys/loginmux_impl.h
161 usr/include/sys/vgareg.h
162 #
163 # Some IPsec headers can't be shipped lest we hit export controls...
164 #
165 usr/include/inet/ipsec_impl.h
166 usr/include/inet/ipsec_info.h
167 usr/include/inet/ipsecah.h
168 usr/include/inet/ipsec esp.h
169 usr/include/inet/keysck.h
170 usr/include/inet/sadb.h
171 usr/include/sys/shal_consts.h
172 usr/include/sys/sha2_consts.h
173 #
174 #
175 # Filtering out directories not shipped
176 #
177 usr/4lib i386
178 #
179 # These files contain definitions shared privately between the kernel
180 # and libc. There is no reason for them to be part of a package that
181 # a customer should ever see. They are installed in the proto area by
182 # the uts build because libc and and other components, like truss, are
183 # dependent upon their contents and should not have their own copies.
184 #
185 usr/include/sys/libc_kernel.h
186 usr/include/sys/synch32.h
187 #
188 # Private interfaces for libdisasm
189 #
190 usr/include/libdisasm.h

```

3

new/exception_lists/packaging

```

191 usr/lib/llib-ldisasm
192 usr/lib/llib-ldisasm.ln
193 usr/lib/amd64/llib-ldisasm.ln i386
194 usr/lib/sparcv9/llib-ldisasm.ln sparc
195 #
196 # Private interfaces for libraidcfg
197 #
198 usr/include/raidcfg_spi.h
199 usr/include/raidcfg.h
200 usr/lib/libraidcfg.so
201 usr/lib/amd64/libraidcfg.so i386
202 usr/lib/sparcv9/libraidcfg.so sparc
203 usr/lib/llib-lraidecfg
204 usr/lib/llib-lraidecfg.ln
205 usr/lib/amd64/llib-lraidecfg.ln i386
206 usr/lib/sparcv9/llib-lraidecfg.ln sparc
207 #
208 # This file is used for private communication between mdb, drv/kmdb, and
209 # misc/kmdb. The interfaces described herein are not intended for customer
210 # use, and are thus excluded from packaging.
211 #
212 usr/include/sys/kmdb.h
213 #
214 # These files are installed in the proto area by the build of libdhcpcagent
215 # and libdhcputil for the benefit of DHCP-related networking commands such
216 # as dhcpcagent, dhcpinfo, ifconfig, and netstat. These are not interfaces
217 # for customer use, so the files are excluded from packaging.
218 #
219 lib/libdhcpcagent.so
220 lib/libdhcputil.so
221 lib/amd64/libdhcputil.so i386
222 lib/sparcv9/libdhcputil.so sparc
223 lib/llib-ldhcpcagent
224 lib/llib-ldhcpcagent.ln
225 lib/llib-ldhcputil
226 lib/llib-ldhcputil.ln
227 lib/amd64/llib-ldhcputil.ln i386
228 lib/sparcv9/llib-ldhcputil.ln sparc
229 usr/include/dhcp_hostconf.h
230 usr/include/dhcp_ impl.h
231 usr/include/dhcp_inittab.h
232 usr/include/dhcp_stable.h
233 usr/include/dhcp_symbol_common.h
234 usr/include/dhcpcagent_ipch.h
235 usr/include/dhcpcagent_util.h
236 usr/include/dhcpcmsg.h
237 usr/lib/libdhcpcagent.so
238 usr/lib/libdhcputil.so
239 usr/lib/amd64/libdhcputil.so i386
240 usr/lib/sparcv9/libdhcputil.so sparc
241 usr/lib/llib-ldhcpcagent
242 usr/lib/llib-ldhcpcagent.ln
243 usr/lib/llib-ldhcputil
244 usr/lib/llib-ldhcputil.ln
245 usr/lib/amd64/llib-ldhcputil.ln i386
246 usr/lib/sparcv9/llib-ldhcputil.ln sparc
247 #
248 # These files are installed in the proto area by the build of libinstzones
249 # and libpkg
250 #
251 usr/lib/llib-linstzones
252 usr/lib/llib-linstzones.ln
253 usr/lib/amd64/llib-linstzones.ln i386
254 usr/lib/sparcv9/llib-linstzones.ln sparc
255 usr/lib/llib-lpkg
256 usr/lib/llib-lpkg.ln

```

4

```

257 #
258 # Don't ship header files private to libipmp and in.mpathd
259 #
260 usr/include/ipmp_query_impl.h
261 #
262 # These files are installed in the proto area by the build of libinetsvc,
263 # an inetd-specific library shared by inetd, inetadm and inetconv. Only
264 # the shared object is shipped.
265 #
266 usr/include/inetsvc.h
267 usr/lib/libinetsvc.so
268 usr/lib/llib-linetsvc
269 usr/lib/llib-linetsvc.ln
270 #
271 # These files are installed in the proto area by the build of libinetutil,
272 # a general purpose library for the benefit of internet utilities. Only
273 # the shared object is shipped.
274 #
275 lib/libinetutil.so
276 lib/amd64/libinetutil.so          i386
277 lib/sparcv9/libinetutil.so       sparc
278 lib/llib-linetutil
279 lib/llib-linetutil.ln
280 lib/amd64/llib-linetutil.ln      i386
281 lib/sparcv9/llib-linetutil.ln    sparc
282 usr/include/libinetutil.h
283 usr/include/netinet/inetutil.h
284 usr/include/ofmt.h
285 usr/lib/libinetutil.so
286 usr/lib/amd64/libinetutil.so     i386
287 usr/lib/sparcv9/libinetutil.so   sparc
288 usr/lib/llib-linetutil
289 usr/lib/llib-linetutil.ln
290 usr/lib/amd64/llib-linetutil.ln  i386
291 usr/lib/sparcv9/llib-linetutil.ln
292 #
293 # Miscellaneous kernel interfaces or kernel<->user interfaces that are
294 # consolidation private and we do not want to export at this time.
295 #
296 usr/include/sys/cryptmod.h
297 usr/include/sys/dumpadm.h
298 usr/include/sys/ontrap.h
299 usr/include/sys/sysmsg_impl.h
300 usr/include/sys/vlan.h
301 #
302 # non-public pci header
303 #
304 usr/include/sys/pci_impl.h
305 usr/include/sys/pci_tools.h
306 #
307 # Exception list for RCM project, included by librcm and rcm_daemon
308 #
309 usr/include/librcm_event.h
310 usr/include/librcm_impl.h
311 #
312 # MDB deliverables that are not yet public
313 #
314 usr/lib/mdb/proc/mdb_test.so
315 usr/lib/mdb/proc/sparcv9/mdb_test.so  sparc
316 #
317 # SNCA project exception list
318 #
319 usr/include/inet/kssl/kssl.h
320 usr/include/inet/kssl/ksslimpl.h
321 usr/include/inet/kssl/ksslproto.h
322 usr/include/inet/nca

```

```

323 #
324 # these are "removed" from the source product build because the only
325 # packages that currently deliver them are removed.
326 # they really should't be in here.
327 #
328 etc/sfw
329 #
330 # Entries for the libmech_krb5 symlink, which has been included
331 # for build purposes only, not delivered to customers.
332 #
333 usr/include/gssapi/gssapi_krb5.h
334 usr/lib/gss/libmech_krb5.so
335 usr/lib/amd64/gss/libmech_krb5.so  i386
336 usr/lib/sparcv9/gss/libmech_krb5.so  sparc
337 usr/lib/libmech_krb5.so
338 usr/lib/amd64/libmech_krb5.so  i386
339 usr/lib/sparcv9/libmech_krb5.so  sparc
340 #
341 # Entries for headers from efcode project which user does not need to see
342 #
343 usr/platform/sun4u/include/sys/fc_plat.h
344 usr/platform/sun4u/include/sys/fcode.h
345 #
346 # Private net80211 headers
347 #
348 usr/include/sys/net80211_amrr.h
349 usr/include/sys/net80211_crypto.h
350 usr/include/sys/net80211_ht.h
351 usr/include/sys/net80211_proto.h
352 usr/include/sys/net80211.h
353 #
354 usr/include/net/wpa.h
355 #
356 # PPPoE files not delivered to customers.
357 #
358 usr/include/net/pppoe.h
359 usr/include/net/sppptun.h
360 #
361 # Simnet
362 #
363 usr/include/net/simnet.h
364 #
365 # Bridging internal data structures
366 #
367 usr/include/net/bridge_impl.h
368 #
369 # User<->kernel interface used by cfgadm/USB only
370 #
371 usr/include/sys/usb/hubd/hubd_impl.h
372 #
373 # User<->kernel interface used by cfgadm/SATA only
374 #
375 usr/include/sys/sata/sata_cfgadm.h
376 #
377 # Private ucred kernel header
378 #
379 usr/include/sys/ucred.h
380 #
381 # Private and/or platform-specific smf(5) files
382 #
383 lib/librestart.so
384 lib/llib-lrestart
385 lib/llib-lrestart.ln
386 lib/amd64/llib-lrestart.ln
387 lib/sparcv9/llib-lrestart.ln
388 usr/include/libcontract_priv.h

```

```

389 usr/include/librestart_priv.h
390 usr/include/librestart.h
391 usr/lib/librestart.so
392 usr/lib/sparcv9/librestart.so
393 lib/svc/manifest/platform/sun4u
394 lib/svc/manifest/platform/sun4v
395 var/svc/manifest/platform/sun4u
396 var/svc/manifest/platform/sun4v
397 etc/svc/profile/platform_sun4v.xml
398 etc/svc/profile/platform_SUNW,SPARC-Enterprise.xml
399 etc/svc/profile/platform_SUNW,Sun-Fire-15000.xml
400 etc/svc/profile/platform_SUNW,Sun-Fire-880.xml
401 etc/svc/profile/platform_SUNW,Sun-Fire-V890.xml
402 etc/svc/profile/platform_SUNW,Sun-Fire.xml
403 etc/svc/profile/platform_SUNW,Ultra-Enterprise-10000.xml
404 etc/svc/profile/platform_SUNW,UltraSPARC-IIe-NetraCT-40.xml
405 etc/svc/profile/platform_SUNW,UltraSPARC-IIe-NetraCT-60.xml
406 etc/svc/profile/platform_SUNW,UltraSPARC-III-Netract.xml
407 #
408 # Private libuutil files
409 #
410 lib/libuutil.so
411 lib/llib-luutil
412 lib/llib-luutil.ln
413 lib/sparcv9/llib-luutil.ln          sparc
414 usr/include/libuutil_impl.h
415 usr/lib/libuutil.so
416 usr/lib/sparcv9/libuutil.so          sparc
417 #
418 # Private Multidata file.
419 #
420 usr/include/sys/multidata_impl.h
421 #
422 # The following files are used by wanboot.
423 # They contain interfaces which are currently private.
424 #
425 usr/include/sys/wanboot_impl.h
426 usr/include/wanboot
427 usr/include/wanbootutil.h
428 #
429 # Even though all the objects built under usr/src/stand are later glommed
430 # together into a couple of second-stage boot loaders, we dump the static
431 # archives and lint libraries into ${ROOT}/stand for intermediate use
432 # (e.g., for lint, linking the second-stage boot loaders, ...). Since
433 # these are merely intermediate objects, they do not need to be packaged.
434 #
435 stand                                sparc
436 #
437 # Private KCF header files
438 #
439 usr/include/sys/crypto/elfsign.h
440 usr/include/sys/crypto/impl.h
441 usr/include/sys/crypto/ops_impl.h
442 usr/include/sys/crypto/sched_impl.h
443 #
444 # The following files are installed in the proto area
445 # by the build of libcmdutils (Command Utilities Library).
446 # libcmdutils contains interfaces which are all private interfaces.
447 #
448 lib/libcmdutils.so
449 lib/amd64/libcmdutils.so          i386
450 lib/sparcv9/libcmdutils.so          sparc
451 lib/llib-lcmdutils
452 lib/llib-lcmdutils.ln
453 lib/amd64/llib-lcmdutils.ln          i386
454 lib/sparcv9/llib-lcmdutils.ln          sparc

```

```

455 usr/include/libcmdutils.h
456 usr/lib/libcmdutils.so
457 usr/lib/amd64/libcmdutils.so          i386
458 usr/lib/sparcv9/libcmdutils.so          sparc
459 usr/lib/llib-lcmdutils
460 usr/lib/llib-lcmdutils.ln
461 usr/lib/amd64/llib-lcmdutils.ln          i386
462 usr/lib/sparcv9/llib-lcmdutils.ln          sparc
463 #
464 # Private interfaces in libsec
465 #
466 usr/include/aclutils.h
467 #
468 # USB skeleton driver stays in sync with the rest of USB but doesn't ship.
469 #
470 kernel/drv/usbskel          i386
471 kernel/drv/amd64/usbskel          i386
472 kernel/drv/sparcv9/usbskel          sparc
473 kernel/drv/usbskel.conf
474 #
475 # Consolidation and Sun private libdevid interfaces
476 # Public libdevid interfaces provided by devid.h
477 #
478 usr/include/sys/libdevid.h
479 #
480 # The following files are installed in the proto area by the build of
481 # libprtdiag. libprtdiag contains interfaces which are all private.
482 # Only the shared object is shipped.
483 #
484 usr/platform/sun4u/lib/llib-lprtdiag          sparc
485 usr/platform/sun4u/lib/llib-lprtdiag.ln          sparc
486 usr/platform/sun4v/lib/llib-lprtdiag.ln          sparc
487 #
488 # The following files are installed in the proto area by the build of
489 # mdesc driver in sun4v. These header files are used on in the build
490 # and do not need to be shipped to customers.
491 #
492 usr/include/sys/mdesc.h
493 usr/include/sys/mdesc_impl.h
494 usr/platform/sun4v/include/sys/mach_descrip.h
495 #
496 # The following files are installed in the proto area by the build of
497 # libpcp. libpcp contains interfaces which are all private.
498 # Only the shared object is shipped.
499 #
500 usr/platform/sun4v/lib/llib-lpcp.ln          sparc
501 usr/platform/SUNW,Netra-CP3060/lib/llib-lpcp.ln          sparc
502 usr/platform/SUNW,Netra-CP3260/lib/llib-lpcp.ln          sparc
503 usr/platform/SUNW,Netra-T5220/lib/llib-lpcp.ln          sparc
504 usr/platform/SUNW,Netra-T5440/lib/llib-lpcp.ln          sparc
505 usr/platform/SUNW,SPARC-Enterprise-T5120/lib/llib-lpcp.ln          sparc
506 usr/platform/SUNW,Sun-Blade-T6300/lib/llib-lpcp.ln          sparc
507 usr/platform/SUNW,Sun-Blade-T6320/lib/llib-lpcp.ln          sparc
508 usr/platform/SUNW,Sun-Fire-T200/lib/llib-lpcp.ln          sparc
509 usr/platform/SUNW,T5140/lib/llib-lpcp.ln          sparc
510 usr/platform/SUNW,USBRDT-5240/lib/llib-lpcp.ln          sparc
511 #
512 # ZFS internal tools and lint libraries
513 #
514 usr/lib/llib-lzfs_jni
515 usr/lib/llib-lzfs_jni.ln
516 usr/lib/amd64/llib-lzfs_jni.ln          i386
517 usr/lib/sparcv9/llib-lzfs_jni.ln          sparc
518 usr/lib/llib-lzpool
519 usr/lib/llib-lzpool.ln          i386
520 usr/lib/amd64/llib-lzpool.ln          i386

```

new/exception_lists/packaging

9

```

521 usr/lib/sparcv9/llib-lzpool.ln      sparc
522 #
523 # ZFS JNI headers
524 #
525 usr/include/libzfs_jni_dataset.h
526 usr/include/libzfs_jni_disk.h
527 usr/include/libzfs_jni_diskmgt.h
528 usr/include/libzfs_jni_ipool.h
529 usr/include/libzfs_jni_main.h
530 usr/include/libzfs_jni_pool.h
531 usr/include/libzfs_jni_property.h
532 usr/include/libzfs_jni_util.h
533 #
534 # These files are installed in the proto area for Solaris scsi_vhci driver
535 # (for MPAPI support) and should not be shipped
536 #
537 usr/include/sys/scsi/adapters/mpapi_impl.h
538 usr/include/sys/scsi/adapters/mpapi_scsi_vhci.h
539 #
540 # This library is installed in the proto area by the build of libdisasm, and is
541 # only used when building the KMDB disasm module.
542 #
543 usr/lib/libstanddisasm.so
544 usr/lib/amd64/libstanddisasm.so      i386
545 usr/lib/sparcv9/libstanddisasm.so    sparc
546 #
547 # TSol: tsol doesn't ship lint source, and tsnet isn't for customers at all.
548 #
549 lib/libtsnet.so
550 usr/lib/llib-ltsnet
551 usr/lib/llib-ltsol
552 #
553 # nss interfaces shared between libnsl and other ON libraries.
554 #
555 usr/include/nss.h
556 #
557 # AT&T AST (ksh93) files which are currently needed only to build OS/Net
558 # (msgcc&co.)
559 # libast
560 usr/lib/libast.so
561 usr/lib/amd64/libast.so            i386
562 usr/lib/sparcv9/libast.so         sparc
563 usr/lib/llib-last
564 usr/lib/llib-last.ln
565 usr/lib/amd64/llib-last.ln        i386
566 usr/lib/sparcv9/llib-last.ln     sparc
567 # libcmd
568 usr/lib/llib-lcmd
569 usr/lib/llib-lcmd.ln
570 usr/lib/amd64/llib-lcmd.ln        i386
571 usr/lib/sparcv9/llib-lcmd.ln     sparc
572 # libdll
573 usr/lib/libdll.so
574 usr/lib/amd64/libdll.so          i386
575 usr/lib/sparcv9/libdll.so        sparc
576 usr/lib/llib-ldll
577 usr/lib/llib-ldll.ln
578 usr/lib/amd64/llib-ldll.ln       i386
579 usr/lib/sparcv9/llib-ldll.ln     sparc
580 # libpp (a helper library needed by AST's msgcc)
581 usr/lib/libpp.so
582 usr/lib/llib-lpp
583 usr/lib/llib-lpp.ln
584 usr/lib/locale/C/LC_MESSAGES/libpp
585 # libshell
586 usr/lib/libshell.so

```

new/exception_lists/packaging

10

```

587 usr/lib/amd64/libshell.so          i386
588 usr/lib/sparcv9/libshell.so        sparc
589 usr/lib/llib-lshell
590 usr/lib/llib-lshell.ln
591 usr/lib/amd64/llib-lshell.ln       i386
592 usr/lib/sparcv9/llib-lshell.ln     sparc
593 # libsum
594 usr/lib/libsum.so
595 usr/lib/amd64/libsum.so           i386
596 usr/lib/sparcv9/libsum.so         sparc
597 usr/lib/llib-lsum
598 usr/lib/llib-lsum.ln
599 usr/lib/amd64/llib-lsum.ln        i386
600 usr/lib/sparcv9/llib-lsum.ln     sparc
601 #
602 # This file is used in ON to build DSCP clients. It is not for customers.
603 #
604 usr/include/libdscp.h             sparc
605 #
606 # These files are used by the iSCSI Target and the iSCSI Initiator
607 #
608 usr/include/sys/iscsi_protocol.h
609 usr/include/sys/iscsi_authclient.h
610 usr/include/sys/iscsi_authclientglue.h
611 #
612 # These files are used by the COMSTAR iSCSI target port provider
613 #
614 usr/include/sys/idm
615 usr/include/sys/iscsit/chap.h
616 usr/include/sys/iscsit/iscsi_if.h
617 usr/include/sys/iscsit/iscns_protocol.h
618 usr/include/sys/iscsit/radius_packet.h
619 usr/include/sys/iscsit/radius_protocol.h
620 #
621 # libshare is private and the 64-bit sharemgr is not delivered.
622 #
623 usr/lib/libshare.so
624 usr/lib/amd64/libshare.so          i386
625 usr/lib/sparcv9/libshare.so        sparc
626 usr/lib/fs/autofs/libshare_automfs.so
627 usr/lib/fs/autofs/amd64/libshare_automfs.so
628 usr/lib/fs/autofs/sparcv9/libshare_automfs.so
629 usr/lib/fs/nfs/libshare_nfs.so
630 usr/lib/fs/nfs/amd64/libshare_nfs.so
631 usr/lib/fs/nfs/sparcv9/libshare_nfs.so
632 usr/lib/fs/nfs/test_svc_tp_create
633 usr/lib/fs/smb/libshare_smb.so
634 usr/lib/fs/smb/amd64/libshare_smb.so
635 usr/lib/fs/smb/sparcv9/libshare_smb.so
636 usr/lib/fs/smbfs/libshare_smbfs.so
637 usr/lib/fs/smbfs/amd64/libshare_smbfs.so
638 usr/lib/fs/smbfs/sparcv9/libshare_smbfs.so
639 usr/include/libshareImpl.h
640 usr/include/scfutil.h
641 #
642 # These files are installed in the proto area by the build of libpri for
643 # the benefit of the builds of FMA libldom, Zeus, picld plugins, and/or
644 # other libpri consumers. However, the libpri interfaces are private to
645 # Sun (Consolidation Private) and not intended for customer use. So these
646 # files (the symlink and the lint library) are excluded from packaging.
647 #
648 usr/lib/libpri.so
649 usr/lib/llib-lpri
650 usr/lib/llib-lpri.ln
651 usr/lib/sparcv9/libpri.so
652 usr/lib/sparcv9/llib-lpri.ln

```

new/exception_lists/packaging

```

653 #
654 # These files are installed in the proto area by the build of libds for
655 # the benefit of the builds of sun4v IO FMA and/or other libds
656 # consumers. However, the libds interfaces are private to Sun
657 # (Consolidation Private) and not intended for customer use. So these
658 # files (the symlink and the lint library) are excluded from packaging.
659 #
660 usr/lib/libds.so          sparc
661 usr/lib/sparcv9/libds.so  sparc
662 usr/lib/l1lib-lds         sparc
663 usr/lib/l1lib-lds.ln      sparc
664 usr/lib/sparcv9/l1lib-lds.ln  sparc
665 usr/lib/libdscfg.so      sparc
666 usr/lib/l1lib-l1dscfg.ln
667 usr/platform/sun4v/include/sys/libds.h  sparc
668 usr/platform/sun4v/include/sys/vlds.h  sparc
669 #
670 # Private/Internal u8_textprep header file. Do not ship.
671 #
672 usr/include/sys/u8_textprep_data.h
673 #
674 # SQLite is private, used by SMF (svc.configd), idmapd and libsmb.
675 #
676 usr/include/sqlite-sys
677 lib/libsqrite-native.o
678 lib/libsqrite-sys.so
679 lib/l1lib-lsqlite-sys
680 lib/l1lib-lsqlite-sys.ln
681 #
682 # Private/Internal kiconv header files. Do not ship.
683 #
684 usr/include/sys/kiconv_big5_utf8.h
685 usr/include/sys/kiconv_cck_common.h
686 usr/include/sys/kiconv_cp950hkscs_utf8.h
687 usr/include/sys/kiconv_emea1.h
688 usr/include/sys/kiconv_emea2.h
689 usr/include/sys/kiconv_euckr_utf8.h
690 usr/include/sys/kiconv_euctw_utf8.h
691 usr/include/sys/kiconv_gb18030_utf8.h
692 usr/include/sys/kiconv_gb2312_utf8.h
693 usr/include/sys/kiconv_hkscs_utf8.h
694 usr/include/sys/kiconv_ja_jis_to_unicode.h
695 usr/include/sys/kiconv_ja_unicode_to_jis.h
696 usr/include/sys/kiconv_ja.h
697 usr/include/sys/kiconv_ko.h
698 usr/include/sys/kiconv_latin1.h
699 usr/include/sys/kiconv_sc.h
700 usr/include/sys/kiconv_tc.h
701 usr/include/sys/kiconv_uhc_utf8.h
702 usr/include/sys/kiconv_utf8_big5.h
703 usr/include/sys/kiconv_utf8_cp950hkscs.h
704 usr/include/sys/kiconv_utf8_euckr.h
705 usr/include/sys/kiconv_utf8_euctw.h
706 usr/include/sys/kiconv_utf8_gb18030.h
707 usr/include/sys/kiconv_utf8_gb2312.h
708 usr/include/sys/kiconv_utf8_hkscs.h
709 usr/include/sys/kiconv_utf8_uhc.h
710 #
711 # At this time, the directory and its contents
712 # are only useful on sun4u systems
713 #
714 etc/flash/postdeployment           i386
715 #
716 # This header file is shared only between the power commands and
717 # ppm/srn modules # and should not be in any package
718 #

```

11

new/exception_lists/packaging

```

719 usr/include/sys/srn.h
720 #
721 # Private/Internal header files of smbsrv. Do not ship.
722 #
723 usr/include/smb
724 usr/include/smbsrv
725 #
726 # Private/Internal libraries for smbsrv in user space
727 #
728 usr/include/libfakekernel
729 usr/lib/libfakekernel.so
730 usr/lib/libfakekernel.so.1
731 usr/lib/l1lib-lfakekernel
732 usr/lib/l1lib-lfakekernel.ln
733 usr/lib/amd64/l1lib-lfakekernel.ln
734 usr/lib/sparcv9/l1lib-lfakekernel.ln
735 usr/lib/amd64/libfakekernel.so
736 usr/lib/amd64/libfakekernel.so.1
737 usr/lib/sparcv9/libfakekernel.so
738 usr/lib/sparcv9/libfakekernel.so.1
739 #
740 # Private/Internal libraries of smbsrv. Do not ship.
741 #
742 usr/lib/mdb/proc/libfksmbsrv.so
743 usr/lib/mdb/proc/amd64/libfksmbsrv.so
744 usr/lib/mdb/proc/sparcv9/libfksmbsrv.so
745 usr/lib/reparse/l1lib-lreparse_smb
746 usr/lib/reparse/l1lib-lreparse_smb.ln
747 usr/lib/smbsrv/bind-helper
748 usr/lib/smbsrv/fksmbsd
749 usr/lib/smbsrv/libfksmbsrv.so
750 usr/lib/smbsrv/libfksmbsrv.so.1
751 usr/lib/smbsrv/l1lib-lfksmbsrv
752 usr/lib/smbsrv/l1lib-lfksmbsrv.ln
753 usr/lib/smbsrv/l1lib-lmlrpc
754 usr/lib/smbsrv/l1lib-lmlrpc.ln
755 usr/lib/smbsrv/l1lib-lmlsvc
756 usr/lib/smbsrv/l1lib-lmlsvc.ln
757 usr/lib/smbsrv/l1lib-lsmb
758 usr/lib/smbsrv/l1lib-lsmb.ln
759 usr/lib/smbsrv/l1lib-lsmbns
760 usr/lib/smbsrv/l1lib-lsmbns.ln
761 #
762 #
763 # Private/Internal 64-bit libraries of smbsrv. Do not ship.
764 #
765 usr/lib/smbsrv/amd64
766 usr/lib/smbsrv/sparcv9
767 #
768 usr/lib/reparse/amd64/libreparse_smb.so
769 usr/lib/reparse/amd64/libreparse_smb.so.1
770 usr/lib/reparse/amd64/l1lib-lreparse_smb.ln
771 usr/lib/reparse/sparcv9/libreparse_smb.so
772 usr/lib/reparse/sparcv9/libreparse_smb.so.1
773 usr/lib/reparse/sparcv9/l1lib-lreparse_smb.ln
774 #
775 # Private dirent, extended to include flags, for use by SMB server
776 #
777 usr/include/sys/extdirent.h
778 #
779 # Private header files for vscan service
780 #
781 usr/include/libvscan.h
782 usr/include/sys/vscan.h
783 #
784 # libvscan is private

```

12

```

785 #
786 usr/lib/vscan/llib-lvscan
787 usr/lib/vscan/llib-lvscan.ln
788 #
789 # i86hvm is not a full platform. It is just a home for paravirtualized
790 # drivers. There is no usr/ component to this sub-platform, but the
791 # directory is created in the proto area to keep other tools happy.
792 #
793 usr/platform/i86hvm                                i386
794 #
795 # Private sdcard framework headers
796 #
797 /usr/include/sys/sdcard
798 #
799 # libssmbfs is private
800 #
801 /usr/include/netsmb
802 /usr/lib/libssmbfs.so
803 /usr/lib/amd64/libssmbfs.so                         i386
804 /usr/lib/sparcv9/libssmbfs.so                      sparc
805 /usr/lib/llib-1smbfs
806 /usr/lib/llib-1smbfs.ln
807 /usr/lib/amd64/llib-1smbfs.ln                       i386
808 /usr/lib/sparcv9/llib-1smbfs.ln                     sparc
809 #
810 # demo & test program for smbfs (private) ACL support
811 #
812 /usr/lib/fs/smbfs/chacl
813 /usr/lib/fs/smbfs/lsacl
814 opt/smbcl-tests
815 #
816 # FC related files
817 kernel/kmdb/fcip                                    i386
818 kernel/kmdb/amd64/fcip                            i386
819 kernel/kmdb/sparcv9/fcip                          sparc
820 kernel/kmdb/fcp                                    i386
821 kernel/kmdb/amd64/fcp                            i386
822 kernel/kmdb/sparcv9/fcp                          sparc
823 kernel/kmdb/fctl                                  i386
824 kernel/kmdb/amd64/fctl                            i386
825 kernel/kmdb/sparcv9/fctl                          sparc
826 kernel/kmdb/qlc                                  i386
827 kernel/kmdb/amd64/qlc                            i386
828 kernel/kmdb/sparcv9/qlc                          sparc
829 lib/llib-la5k                                    sparc
830 lib/llib-la5k.ln                                 sparc
831 lib/sparcv9/llib-la5k.ln                         sparc
832 lib/llib-lg_fc                                  sparc
833 lib/llib-lg_fc.ln                               sparc
834 lib/sparcv9/llib-lg_fc.ln                        sparc
835 /usr/include/a_state.h                           sparc
836 /usr/include/a5k.h                               sparc
837 /usr/include/exec.h                            sparc
838 /usr/include/g_scsi.h                           sparc
839 /usr/include/g_state.h                           sparc
840 /usr/include/gfc.h                             sparc
841 /usr/include/l_common.h                           sparc
842 /usr/include/l_error.h                           sparc
843 /usr/include/rom.h                             sparc
844 /usr/include/stgcom.h                           sparc
845 /usr/include/sys/fibre-channel
846 /usr/lib/llib-lHBAPI
847 /usr/lib/llib-lHBAPI.ln
848 /usr/lib/amd64/llib-lHBAPI.ln                  i386
849 /usr/lib/sparcv9/llib-lHBAPI.ln                 sparc
850 #

```

```

851 /usr/bin/dscfgcli
852 /usr/bin/sd_diag
853 /usr/bin/sd_stats
854 /usr/include/nsctl.h
855 /usr/include/sys/ncall
856 /usr/include/sys/nsc_ddi.h
857 /usr/include/sys/nsc_thread.h
858 /usr/include/sys/nsctl
859 /usr/include/sys/nskernd.h
860 /usr/include/sys/unistat
861 /usr/lib/libnsctl.so
862 /usr/lib/librfdc.so
863 /usr/lib/libunistat.so
864 /usr/lib/llib-1nsctl.ln
865 /usr/lib/llib-lrdc.ln
866 /usr/lib/llib-lunistat.ln
867 #
868 # These files are used by the iSCSI initiator only.
869 # No reason to ship them.
870 #
871 /usr/include/sys/scsi/adapters/iscsi_door.h
872 /usr/include/sys/scsi/adapters/iscsi_if.h
873 #
874 # sbd ioctl hdr
875 #
876 /usr/include/sys/stmf_sbd_ioctl.h
877 #
878 # proxy port provider interface
879 #
880 /usr/include/sys/pppt_ic_if.h
881 /usr/include/sys/pppt_ioctl.h
882 #
883 # proxy daemon lint library
884 #
885 /usr/lib/llib-lstmfproxy
886 /usr/lib/llib-lstmfproxy.ln
887 /usr/lib/amd64/llib-lstmfproxy.ln                i386
888 /usr/lib/sparcv9/llib-lstmfproxy.ln             sparc
889 #
890 # portable object file and dictionary used by libfmd_msg test
891 #
892 /usr/lib/fm/dict/TEST.dict
893 /usr/lib/locale/C/LC_MESSAGES/TEST.mo
894 /usr/lib/locale/C/LC_MESSAGES/TEST.po
895 #
896 # Private idmap RPC protocol
897 #
898 /usr/include/rpcsvc/idmap_prot.h
899 /usr/include/rpcsvc/idmap_prot.x
900 #
901 # Private idmap directory API
902 #
903 /usr/include/directory.h
904 #
905 # librstp is private for bridging
906 #
907 /usr/include/stp_bpdu.h
908 /usr/include/stp_in.h
909 /usr/include/stp_vectors.h
910 /usr/lib/librstp.so
911 /usr/lib/llib-lrstp
912 /usr/lib/llib-lrstp.ln
913 #
914 # Private nvfru API
915 #
916 /usr/include/nvfru.h

```

```

917 #
918 # vrrp
919 #
920 usr/include/libvrrpadm.h
921 usr/lib/libvrrpadm.so
922 usr/lib/amd64/libvrrpadm.so      i386
923 usr/lib/sparcv9/libvrrpadm.so    sparc
924 usr/lib/llib-lvrrpadm
925 usr/lib/llib-lvrrpadm.ln
926 usr/lib/amd64/llib-lvrrpadm.ln   i386
927 usr/lib/sparcv9/llib-lvrrpadm.ln sparc
928 #
929 # This is only used during the -t tools build
930 #
931 opt/onbld/bin/i386/mandoc      i386
932 opt/onbld/bin/sparc/mandoc     sparc
934 #
935 # Private libdwarf
936 #
937 opt/onbld/lib/i386/libdwarf.so  i386
938 opt/onbld/lib/sparc/libdwarf.so sparc
940 #
941 # Private socket filter API
942 #
943 usr/include/sys/sockfilter.h
944 #
945 # We don't actually validate license action payloads, and the license
946 # staging area is provided as a separate basedir for package
947 # publication. The net result is that everything therein should be
948 # ignored for packaging validation.
949 #
950 licenses
951 #
952 # Libbe is private
953 #
954 usr/include/libbe_priv.h
955 #
956 # ipmi is at present only useful on i386, but for historical reasons is
957 # delivered on SPARC and used by the build.
958 #
959 usr/include/sys/ipmi.h  sparc
961 #
962 # libsaveargs is private
963 #
964 usr/include/saveargs.h          i386
965 usr/lib/amd64/libsaveargs.so    i386
966 usr/lib/amd64/libstandsavargs.so i386
967 usr/lib/amd64/llib-lsaveargs.ln i386
969 #
970 # libpcidb is private
971 #
972 usr/include/pcidb.h
973 usr/lib/amd64/libpcidb.so       i386
974 usr/lib/amd64/llib-lpcidb.ln    i386
975 usr/lib/sparcv9/libpcidb.so     sparc
976 usr/lib/sparcv9/llib-lpcidb.ln  sparc
977 usr/lib/libpcidb.so
978 usr/lib/llib-lpcidb
979 usr/lib/llib-lpcidb.ln
981 #
982 # private nvme header file

```

```

983 #
984 usr/include/sys/nvme.h
986 #
987 # debugging program for libadutils
988 #
989 usr/bin/test-getdc
990 #
991 # libflicl-sys is private
992 #
993 usr/include/ficllocal.h
994 usr/lib/amd64/llib-lficl-sys.ln  i386
995 usr/lib/amd64/libflicl-sys.so    i386
996 usr/lib/sparcv9/llib-lficl-sys.ln sparc
997 usr/lib/sparcv9/libflicl-sys.so  sparc
998 usr/lib/llib-lficl-sys
999 usr/lib/llib-lficl-sys.ln
1000 usr/lib/libflicl-sys.so

```

new/usr/src/cmd/fs.d/nfs/Makefile

```
*****
1807 Mon Jul 17 16:15:15 2017
new/usr/src/cmd/fs.d/nfs/Makefile
8330 Add svc_tp_create_addr to libns1
Reviewed by: Paul Dagnelie <pcd@delphix.com>
Reviewed by: Evan Layton <evan.layton@nexenta.com>
Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # cmd/fs.d/nfs/Makefile
27 #
28 # cmd/fs.d/nfs is the directory of all nfs specific commands
29 # whose executable reside in $(INSDIR1) and $(INSDIR2).
30 #

32 include $(SRC)/Makefile.master

34 SUBDIR1=
35         exportfs nfsvd rquotad \
36         statd nfsstat mountd dfshares \
36         nfsfind nfs4cbd share tests
36         nfsfind nfs4cbd share

38 # These do "make catalog"
39 SUBDIR2=
40         clear_locks lockd umount showmount \
41         mount dfmounts nfslog nfsmapid \
41         nfsref rp_basic

43 SUBDIR3=
44 SUBDIRS=
44         $(SUBDIR1) $(SUBDIR2) $(SUBDIR3)

46 # for messaging catalog files
47 #
48 POFILES=
48         $(SUBDIR2:%%/%.po)
49 POFILE=
49         nfs.po

51 all:=          TARGET= all
52 install:=      TARGET= install
53 clean:=        TARGET= clean
54 clobber:=      TARGET= clobber
55 lint:=         TARGET= lint
56 catalog:=     TARGET= catalog
```

1

new/usr/src/cmd/fs.d/nfs/Makefile

```
58 .KEEP_STATE:
60 .PARALLEL:      $(SUBDIRS)
62 all install clean clobber lint: $(SUBDIRS)
64 catalog: $(SUBDIR2)
65         $(RM) $(POFILE)
66         cat $(POFILES) > $(POFILE)
68 $(SUBDIRS): FRC
69         @cd $@; pwd; $(MAKE) $(TARGET)
71 FRC:
```

2

new/usr/src/cmd/fs.d/nfs/mountd/mountd.c

73640 Mon Jul 17 16:15:16 2017

new/usr/src/cmd/fs.d/nfs/mountd/mountd.c

7577 mountd support to run on a fixed port

Portions contributed by: Paul Dagnelie <pcd@delphix.com>

Reviewed by: Evan Layton <evan.layton@nexenta.com>

Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>

```
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 2015 Nexenta Systems, Inc. All rights reserved.
24 * Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
25 * Copyright (c) 2012, 2016 by Delphix. All rights reserved.
26 */
27
28 /* Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989 AT&T */
29 /* All Rights Reserved */
30
31 /*
32 * Portions of this source code were derived from Berkeley 4.3 BSD
33 * under license from the Regents of the University of California.
34 */

35
36 #include <stdio.h>
37 #include <stdio_ext.h>
38 #include <stdlib.h>
39 #include <ctype.h>
40 #include <sys/types.h>
41 #include <string.h>
42 #include <syslog.h>
43 #include <sys/param.h>
44 #include <rpc/rpc.h>
45 #include <sys/stat.h>
46 #include <netconfig.h>
47 #include <netdir.h>
48 #include <sys/file.h>
49 #include <sys/time.h>
50 #include <sys/errno.h>
51 #include <rpccsvc/mount.h>
52 #include <sys/pathconf.h>
53 #include <sys/systeminfo.h>
54 #include <sys/utsname.h>
55 #include <sys/wait.h>
56 #include <sys/resource.h>
57 #include <signal.h>
```

1

new/usr/src/cmd/fs.d/nfs/mountd/mountd.c

```
58 #include <locale.h>
59 #include <unistd.h>
60 #include <errno.h>
61 #include <sys/socket.h>
62 #include <netinet/in.h>
63 #include <arpa/inet.h>
64 #include <netdb.h>
65 #include <thread.h>
66 #include <assert.h>
67 #include <priv_utils.h>
68 #include <nfs/auth.h>
69 #include <nfs/nfssys.h>
70 #include <nfs/nfs.h>
71 #include <nfs/nfs_sec.h>
72 #include <rpccsvc/daemon_utils.h>
73 #include <defl.h>
74 #include "../fslib.h"
75 #include <sharefs/share.h>
76 #include <sharefs/sharetab.h>
77 #include "../lib/sharetab.h"
78 #include "mountd.h"
79 #include <tsol/label.h>
80 #include <sys/tsol/label_macro.h>
81 #include <libtsnet.h>
82 #include <sys/sdt.h>
83 #include <libscf.h>
84 #include <limits.h>
85 #include <sys/nvpair.h>
86 #include <attr.h>
87 #include "smfcfg.h"
88 #include <pwd.h>
89 #include <grp.h>
90 #include <alloca.h>

92 extern int daemonize_init(void);
93 extern void daemonize_fini(int);

95 extern int _nfssys(int, void *);

97 struct sh_list *share_list;

99 rwlock_t sharetab_lock; /* lock to protect the cached sharetab */
100 static mutex_t mnttab_lock; /* prevent concurrent mnttab readers */

102 static mutex_t logging_queue_lock;
103 static cond_t logging_queue_cv;

105 static share_t *find_lofsentry(char *, int *);
106 static int getclientsflavors_old(share_t *, struct cln *, int *);
107 static int getclientsflavors_new(share_t *, struct cln *, int *);
108 static int check_client_old(share_t *, struct cln *, int, uid_t, gid_t, uint_t,
109     gid_t *, uid_t *, gid_t *, uint_t *, gid_t **);
110 static int check_client_new(share_t *, struct cln *, int, uid_t, gid_t, uint_t,
111     gid_t *, uid_t *, gid_t *i, uint_t *, gid_t **);
112 static void mnt(struct svc_req *, SVCXPRT *);
113 static void mnt_pathconf(struct svc_req *);
114 static int mount(struct svc_req *);
115 static void sh_free(struct sh_list *);
116 static void umount(struct svc_req *);
117 static void umountall(struct svc_req *);
118 static int newopts(char *);
119 static tsol_tpent_t *get_client_template(struct sockaddr *);

121 static int debug;
122 static int verbose;
123 static int rejecting;
```

2

```

124 static int mount_vers_min = MOUNTVERS;
125 static int mount_vers_max = MOUNTVERS3;
126 static int mountd_port = 0;

128 extern void nfscmd_func(void *, char *, size_t, door_desc_t *, uint_t);

130 thread_t      nfsauth_thread;
131 thread_t      cmd_thread;
132 thread_t      logging_thread;

134 typedef struct logging_data {
135     char          *ld_host;
136     char          *ld_path;
137     char          *ld_rpath;
138     int           ld_status;
139     char          *ld_netid;
140     struct netbuf *ld_nb;
141     struct logging_data *ld_next;
142 } logging_data;
unchanged portion omitted

368 /*
369  * This function is called for each configured network type to
370  * bind and register our RPC service programs.
371  *
372  * On TCP or UDP, we may want to bind MOUNTPROG on a specific port
373  * (when mountd_port is specified) in which case we'll use the
374  * variant of svc_tp_create() that lets us pass a bind address.
375  */
376 static void
377 md_svc_tp_create(struct netconfig *nconf)
378 {
379     char port_str[8];
380     struct nd_hostserv hs;
381     struct nd_addrlist *al = NULL;
382     SVCXPRT *xpvt = NULL;
383     rpcvers_t vers;
384
385     vers = mount_vers_max;
386
387     /*
388      * If mountd_port is set and this is an inet transport,
389      * bind this service on the specified port.  The TLI way
390      * to create such a bind address is netdir_getbyname()
391      * with the special "host" HOST_SELF_BIND.  This builds
392      * an all-zeros IP address with the specified port.
393      */
394     if (mountd_port != 0 &&
395         (strcmp(nconf->nc_protofmy, NC_INET) == 0 ||
396          strcmp(nconf->nc_protofmy, NC_INET6) == 0)) {
397         int err;
398
399         sprintf(port_str, sizeof (port_str), "%u",
400                 (unsigned short)mountd_port);
401
402         hs.h_host = HOST_SELF_BIND;
403         hs.h_serv = port_str;
404         err = netdir_getbyname((struct netconfig *)nconf, &hs, &al);
405         if (err == 0 && al != NULL) {
406             xpvt = svc_tp_create_addr(mnt, MOUNTPROG, vers,
407                                       nconf, al->n_addrs);
408             netdir_free(al, ND_ADDRLIST);
409         }
410         if (xpvt == NULL) {
411             syslog(LOG_ERR, "mountd: unable to create "
412                   "(MOUNTD,%d) on transport %s (port %d)",
```

```

413                                         vers, nconf->nc_netid, mountd_port);
414
415     /* fall-back to default bind */
416 }
417 if (xpvt == NULL) {
418     /*
419      * Had mountd_port=0, or non-inet transport,
420      * or the bind to a specific port failed.
421      * Do a default bind.
422      */
423     xpvt = svc_tp_create(mnt, MOUNTPROG, vers, nconf);
424
425     if (xpvt == NULL) {
426         syslog(LOG_ERR, "mountd: unable to create "
427               "(MOUNTD,%d) on transport %s",
428               vers, nconf->nc_netid);
429         return;
430     }
431
432     /*
433      * Register additional versions on this transport.
434      */
435     while (--vers >= mount_vers_min) {
436         if (!svc_reg(xpvt, MOUNTPROG, vers, mnt, nconf)) {
437             (void) syslog(LOG_ERR, "mountd: "
438                         "failed to register vers %d on %s",
439                         vers, nconf->nc_netid);
440         }
441     }
442 }

443 int
444 main(int argc, char *argv[])
445 {
446     int      pid;
447     int      c;
448     int      rpc_svc_fdunlm = 1;
449     int      rpc_svc_mode = RPC_SVC_MT_AUTO;
450     int      maxrecsz = RPC_MAXDATASIZE;
451     bool_t   exclbind = TRUE;
452     bool_t   can_do_mlp;
453     long    thr_flags = (THR_NEW_LWP | THR_DAEMON);
454     char    defval[4];
455     int      defvers, ret, bufsz;
456     struct rlimit rl;
457     int      listen_backlog = 0;
458     int      max_threads = 0;
459     int      tmp;
460
461     struct netconfig *nconf;
462     NCONF_HANDLE *nc;
```

463 pipe_fd = -1;

464 /*
465 * Mountd requires uid 0 for:
466 * /etc/rmtab updates (we could chown it to daemon)
467 * /etc/dfs/dfstab reading (it wants to lock out share which
468 * doesn't do any locking before first truncate;
469 * NFS share does; should use fcntl locking instead)
470 * Needed privileges:
471 * auditing
472 * nfs syscall
473 * file dac search (so it can stat all files)
474 * Optional privileges:
475 * MLP
476 */
477 }

```

479     can_do_mlp = priv_ineffect(PRIV_NET_BINDMLP);
480     if (_init_daemon_priv(PU_RESETGROUPS|PU_CLEARLIMITSET, -1, -1,
481         PRIV_SYS_NFS, PRIV_PROC_AUDIT, PRIV_FILE_DAC_SEARCH,
482         PRIV_NET_PRIVADDR,
483         can_do_mlp ? PRIV_NET_BINDMLP : NULL, NULL) == -1) {
484         (void) fprintf(stderr,
485             "%s: must be run with sufficient privileges\n",
486             argv[0]);
487         exit(1);
488     }
489
490     if (getrlimit(RLIMIT_NOFILE, &rl) != 0) {
491         syslog(LOG_ERR, "getrlimit failed");
492     } else {
493         rl.rlim_cur = rl.rlim_max;
494         if (setrlimit(RLIMIT_NOFILE, &rl) != 0)
495             syslog(LOG_ERR, "setrlimit failed");
496     }
497
498     (void) enable_extended_FILE_stdio(-1, -1);
499
500     ret = nfs_smf_get_iprop("mountd_max_threads", &max_threads,
501         DEFAULT_INSTANCE, SCF_TYPE_INTEGER, NFSD);
502     if (ret != SA_OK) {
503         syslog(LOG_ERR, "Reading of mountd_max_threads from SMF "
504             "failed, using default value");
505     }
506
507     ret = nfs_smf_get_iprop("mountd_port", &mountd_port,
508         DEFAULT_INSTANCE, SCF_TYPE_INTEGER, NFSD);
509     if (ret != SA_OK) {
510         syslog(LOG_ERR, "Reading of mountd_port from SMF "
511             "failed, using default value");
512     }
513
514     while ((c = getopt(argc, argv, "dvrm:p:")) != EOF) {
515         while ((c = getopt(argc, argv, "vrm:")) != EOF) {
516             switch (c) {
517                 case 'd':
518                     debug++;
519                     break;
520                 case 'v':
521                     verbose++;
522                     break;
523                 case 'r':
524                     rejecting = 1;
525                     break;
526                 case 'm':
527                     if (convert_int(&tmp, optarg) != 0 || tmp < 1) {
528                         (void) fprintf(stderr, "%s: invalid "
529                             "max_threads option, using defaults\n",
530                             argv[0]);
531                     break;
532                 }
533                 max_threads = tmp;
534                 break;
535                 case 'p':
536                     if (convert_int(&tmp, optarg) != 0 || tmp < 1 ||
537                         tmp > UINT16_MAX) {
538                         (void) fprintf(stderr, "%s: invalid port "
539                             "number\n", argv[0]);
540                     break;
541                 }
542                 mountd_port = tmp;
543                 break;
544             default:

```

```

544             fprintf(stderr, "usage: mountd [-v] [-r]\n");
545             exit(1);
546         }
547     }
548
549     /*
550      * Read in the NFS version values from config file.
551      */
552     bufsz = 4;
553     ret = nfs_smf_get_prop("server_versmin", defval, DEFAULT_INSTANCE,
554         SCF_TYPE_INTEGER, NFSD, &bufsz);
555     if (ret == SA_OK) {
556         errno = 0;
557         defvers = strtol(defval, (char **)NULL, 10);
558         if (errno == 0) {
559             mount_vers_min = defvers;
560             /*
561              * special because NFSv2 is
562              * supported by mount v1 & v2
563              */
564             if (defvers == NFS_VERSION)
565                 mount_vers_min = MOUNTVERS;
566         }
567     }
568
569     bufsz = 4;
570     ret = nfs_smf_get_prop("server_versmax", defval, DEFAULT_INSTANCE,
571         SCF_TYPE_INTEGER, NFSD, &bufsz);
572     if (ret == SA_OK) {
573         errno = 0;
574         defvers = strtol(defval, (char **)NULL, 10);
575         if (errno == 0) {
576             mount_vers_max = defvers;
577         }
578     }
579
580     ret = nfs_smf_get_iprop("mountd_listen_backlog", &listen_backlog,
581         DEFAULT_INSTANCE, SCF_TYPE_INTEGER, NFSD);
582     if (ret != SA_OK) {
583         syslog(LOG_ERR, "Reading of mountd_listen_backlog from SMF "
584             "failed, using default value");
585     }
586
587     /*
588      * Sanity check versions,
589      * even though we may get versions > MOUNTVERS3, we still need
590      * to start nfsauth service, so continue on regardless of values.
591      */
592     if (mount_vers_max > MOUNTVERS3)
593         mount_vers_max = MOUNTVERS3;
594     if (mount_vers_min > mount_vers_max) {
595         fprintf(stderr, "server_versmin > server_versmax\n");
596         mount_vers_max = mount_vers_min;
597     }
598     (void) setlocale(LC_ALL, "");
599     (void) rwlock_init(&sharetab_lock, USYNC_THREAD, NULL);
600     (void) mutex_init(&mnttab_lock, USYNC_THREAD, NULL);
601     (void) mutex_init(&logging_queue_lock, USYNC_THREAD, NULL);
602     (void) cond_init(&logging_queue_cv, USYNC_THREAD, NULL);
603
604     netgroup_init();
605
606 #if !defined(TEXT_DOMAIN)
607 #define TEXT_DOMAIN "SYS_TEST"
608 #endif
609     (void) textdomain(TEXT_DOMAIN);

```

```

611     /* Don't drop core if the NFS module isn't loaded. */
612     (void) signal(SIGSYS, SIG_IGN);
614     if (!debug)
615         pipe_fd = daemonize_init();
617     /*
618      * If we coredump it'll be in /core
619      */
620     if (chdir("/") < 0)
621         fprintf(stderr, "chdir /: %s\n", strerror(errno));
623     if (!debug)
624         openlog("mountd", LOG_PID, LOG_DAEMON);
626     /*
627      * establish our lock on the lock file and write our pid to it.
628      * exit if some other process holds the lock, or if there's any
629      * error in writing/locking the file.
630      */
631     pid = _enter_daemon_lock(MOUNTD);
632     switch (pid) {
633     case 0:
634         break;
635     case -1:
636         fprintf(stderr, "error locking for %s: %s\n", MOUNTD,
637                 strerror(errno));
638         exit(2);
639     default:
640         /* daemon was already running */
641         exit(0);
642     }
644     audit_mountd_setup(); /* BSM */
646     /*
647      * Get required system variables
648      */
649     if ((ngroups_max = sysconf(_SC_NGROUPS_MAX)) == -1) {
650         syslog(LOG_ERR, "Unable to get _SC_NGROUPS_MAX");
651         exit(1);
652     }
653     if ((pw_size = sysconf(_SC_GETPW_R_SIZE_MAX)) == -1) {
654         syslog(LOG_ERR, "Unable to get _SC_GETPW_R_SIZE_MAX");
655         exit(1);
656     }
658     /*
659      * Set number of file descriptors to unlimited
660      */
661     if (!rpc_control(RPC_SVC_USE_POLLFD, &rpc_svc_fdunlim)) {
662         syslog(LOG_INFO, "unable to set number of FDs to unlimited");
663     }
665     /*
666      * Tell RPC that we want automatic thread mode.
667      * A new thread will be spawned for each request.
668      */
669     if (!rpc_control(RPC_SVC_MTMODE_SET, &rpc_svc_mode)) {
670         fprintf(stderr, "unable to set automatic MT mode\n");
671         exit(1);
672     }
674     /*
675      * Enable non-blocking mode and maximum record size checks for

```

```

676             * connection oriented transports.
677             */
678             if (!rpc_control(RPC_SVC_CONNMAXREC_SET, &maxrecsz)) {
679                 fprintf(stderr, "unable to set RPC max record size\n");
680             }
682             /*
683             * Prevent our non-priv udp and tcp ports bound w/wildcard addr
684             * from being hijacked by a bind to a more specific addr.
685             */
686             if (!rpc_control(__RPC_SVC_EXCLBIND_SET, &exclbind)) {
687                 fprintf(stderr, "warning: unable to set udp/tcp EXCLBIND\n");
688             }
690             /*
691             * Set the maximum number of outstanding connection
692             * indications (listen backlog) to the value specified.
693             */
694             if (listen_backlog > 0 && !rpc_control(__RPC_SVC_LSTNBKLOG_SET,
695                 &listen_backlog)) {
696                 fprintf(stderr, "unable to set listen backlog\n");
697                 exit(1);
698             }
700             /*
701             * If max_threads was specified, then set the
702             * maximum number of threads to the value specified.
703             */
704             if (max_threads > 0 && !rpc_control(RPC_SVC_THRMAX_SET, &max_threads)) {
705                 fprintf(stderr, "unable to set max_threads\n");
706                 exit(1);
707             }
709             if (mounted_port < 0 || mounted_port > UINT16_MAX) {
710                 fprintf(stderr, "unable to use specified port\n");
711                 exit(1);
712             }
714             /*
715             * Make sure to unregister any previous versions in case the
716             * user is reconfiguring the server in interesting ways.
717             */
718             svc_unreg(MOUNTPROG, MOUNTVERS);
719             svc_unreg(MOUNTPROG, MOUNTVERS_POSIX);
720             svc_unreg(MOUNTPROG, MOUNTVERS3);
722             /*
723             * Create the nfsauth thread with same signal disposition
724             * as the main thread. We need to create a separate thread
725             * since mountd() will be both an RPC server (for remote
726             * traffic) and a doors server (for kernel upcalls).
727             */
728             if (thr_create(NULL, 0, nfsauth_svc, 0, thr_flags, &nfsauth_thread)) {
729                 fprintf(stderr,
730                         gettext("Failed to create NFSAUTH svc thread\n"));
731                 exit(2);
732             }
734             /*
735             * Create the cmd service thread with same signal disposition
736             * as the main thread. We need to create a separate thread
737             * since mountd() will be both an RPC server (for remote
738             * traffic) and a doors server (for kernel upcalls).
739             */
740             if (thr_create(NULL, 0, cmd_svc, 0, thr_flags, &cmd_thread)) {
741                 syslog(LOG_ERR, gettext("Failed to create CMD svc thread"));

```

```

742         exit(2);
743     }
744
745     /* Create an additional thread to service the rmtab and
746      * audit_mountd_mount logging for mount requests. Use the same
747      * signal disposition as the main thread. We create
748      * a separate thread to allow the mount request threads to
749      * clear as soon as possible.
750 */
751 if (thr_create(NULL, 0, logging_svc, 0, thr_flags, &logging_thread)) {
752     syslog(LOG_ERR, gettext("Failed to create LOGGING svc thread"));
753     exit(2);
754 }
755
756 /*
757  * Enumerate network transports and create service listeners
758  * as appropriate for each.
759  * Create datagram and connection oriented services
760  */
761 if ((nc = setnetconfig()) == NULL) {
762     syslog(LOG_ERR, "setnetconfig failed: %m");
763     return (-1);
764 if (mount_ver_max >= MOUNTVERS) {
765     if (svc_create(mnt, MOUNTPROG, MOUNTVERS, "datagram_v") == 0) {
766         fprintf(stderr,
767                 "couldn't register datagram_v MOUNTVERS\n");
768         exit(1);
769     }
770     while ((nconf = getnetconfig(nc)) != NULL) {
771         /*
772          * Skip things like tpi_raw, invisible...
773          */
774         if ((nconf->nc_flag & NC_VISIBLE) == 0)
775             continue;
776         if (nconf->nc_semantics != NC_TPI_CLTS &&
777             nconf->nc_semantics != NC_TPI_COTS &&
778             nconf->nc_semantics != NC_TPI_COTS_ORD)
779             continue;
780         if (svc_create(mnt, MOUNTPROG, MOUNTVERS, "circuit_v") == 0) {
781             fprintf(stderr,
782                     "couldn't register circuit_v MOUNTVERS\n");
783             exit(1);
784         }
785     }
786
787     md_svc_tp_create(nconf);
788     if (mount_ver_max >= MOUNTVERS_POSIX) {
789         if (svc_create(mnt, MOUNTPROG, MOUNTVERS_POSIX,
790                     "datagram_v") == 0) {
791             fprintf(stderr,
792                     "couldn't register datagram_v MOUNTVERS_POSIX\n");
793             exit(1);
794     }
795     void) endnetconfig(nc);
796     if (svc_create(mnt, MOUNTPROG, MOUNTVERS_POSIX,
797                 "circuit_v") == 0) {
798         fprintf(stderr,
799                 "couldn't register circuit_v MOUNTVERS_POSIX\n");
800         exit(1);
801     }
802
803     if (mount_ver_max >= MOUNTVERS3) {
804         if (svc_create(mnt, MOUNTPROG, MOUNTVERS3, "datagram_v") == 0) {
805             fprintf(stderr,
806

```

```

807                 "couldn't register datagram_v MOUNTVERS3\n");
808         }
809     }
810     if (svc_create(mnt, MOUNTPROG, MOUNTVERS3, "circuit_v") == 0) {
811         fprintf(stderr,
812                 "couldn't register circuit_v MOUNTVERS3\n");
813         exit(1);
814     }
815
816     /*
817      * Start serving
818      */
819     rmtab_load();
820
821     daemonize_fini(pipe_fd);
822
823     /* Get rid of the most dangerous basic privileges. */
824     __fini_daemon_priv(PRIV_PROC_EXEC, PRIV_PROC_INFO, PRIV_PROC_SESSION,
825                         (char *)NULL);
826
827     svc_run();
828     syslog(LOG_ERR, "Error: svc_run shouldn't have returned");
829     abort();
830
831     /* NOTREACHED */
832     return (0);
833 }
834
835 unchanged_portion_omitted

```

new/usr/src/cmd/fs.d/nfs/statd/Makefile

```
*****
1875 Mon Jul 17 16:15:16 2017
new/usr/src/cmd/fs.d/nfs/statd/Makefile
7569 statd support to run on a fixed port
Portions contributed by: Paul Dagnelie <pcd@delphix.com>
Reviewed by: Evan Layton <evan.layton@nexenta.com>
Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>
*****
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, Version 1.0 only
6 # (the "License"). You may not use this file except in compliance
7 # with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #

23 #
24 # Copyright 2015 Nexenta Systems, Inc. All rights reserved.
25 #

27 #
28 # Copyright 1990-2003 Sun Microsystems, Inc. All rights reserved.
29 # Use is subject to license terms.
30 #
31 # Copyright (c) 2016 by Delphix. All rights reserved.
32 #

34 FSTYPE= nfs
35 TYPEPROG= statd
36 ATTMK= $(TYPEPROG)

38 include ../../Makefile.fstype
39 CPPFLAGS += -D_REENTRANT -DSUN_THREADS

41 CERRWARN += -_gcc=-Wno-unused-variable
42 CERRWARN += -_gcc=-Wno-switch
43 CERRWARN += -_gcc=-Wno-parentheses
44 CERRWARN += -_gcc=-Wno-uninitialized

46 LOCAL= sm_svc.o sm_proc.o sm_statd.o
47 OJBJS= $(LOCAL) selfcheck.o daemon.o smfcfg.o
44 OJBJS= $(LOCAL) selfcheck.o daemon.o

49 SRCS= $(LOCAL:.o=%.c) ./lib/selfcheck.c ./lib/daemon.c \
50 ./lib/smfcfg.c
46 SRCS= $(LOCAL:.o=% .c) ./lib/selfcheck.c ./lib/daemon.c

52 LDLIBS += -lsocket -lrpcsvc -lnsl -lscf
53 CPPFLAGS += -I./lib
48 LDLIBS += -lsocket -lrpcsvc -lnsl

55 $(TYPEPROG): $(OJBJS)
```

1

new/usr/src/cmd/fs.d/nfs/statd/Makefile

```
56      $(LINK.C) -o $@ $(OJBJS) $(LDLIBS)
57      $(POST_PROCESS)
58      $(LOCK_LINT)

60 selfcheck.o: ./lib/selfcheck.c
61      $(COMPILE.C) ./lib/selfcheck.c

63 daemon.o: ./lib/daemon.c
64      $(COMPILE.C) ./lib/daemon.c

66 smfcfg.o: ./lib/smfcfg.c
67      $(COMPILE.C) ./lib/smfcfg.c

69 lint: lint_SRCS

71 clean:
72      $(RM) $(OJBJS) $(TYPEPROG)
```

2

new/usr/src/cmd/fs.d/nfs/statd/sm_svc.c

24610 Mon Jul 17 16:15:16 2017

new/usr/src/cmd/fs.d/nfs/statd/sm_svc.c

7569 statd support to run on a fixed port

Portions contributed by: Paul Dagnelie <pcd@delphix.com>

Reviewed by: Evan Layton <evan.layton@nexenta.com>

Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>

```
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 2015 Nexenta Systems, Inc. All rights reserved.
23 * Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
24 * Copyright (c) 2012, 2016 by Delphix. All rights reserved.
25 * Copyright 2016 Nexenta Systems, Inc. All rights reserved.
26 */
```

```
28 /* Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
29 /* All Rights Reserved */
```

```
31 /*
32 * University Copyright- Copyright (c) 1982, 1986, 1988
33 * The Regents of the University of California
34 * All Rights Reserved
35 *
36 * University Acknowledgment- Portions of this document are derived from
37 * software developed by the University of California, Berkeley, and its
38 * contributors.
39 */
```

```
40 /*
41 * Copyright (c) 2012 by Delphix. All rights reserved.
42 */
```

```
41 #include <stdio.h>
42 #include <stdio_ext.h>
43 #include <stdlib.h>
44 #include <ftw.h>
45 #include <signal.h>
46 #include <string.h>
47 #include <syslog.h>
48 #include <netconfig.h>
49 #include <netdir.h>
50 #include <unistd.h>
51 #include <netdb.h>
52 #include <rpc/rpc.h>
53 #include <rpc/svc.h>
```

1

new/usr/src/cmd/fs.d/nfs/statd/sm_svc.c

```
54 #include <netinet/in.h>
55 #include <sys/param.h>
56 #include <sys/resource.h>
57 #include <sys/file.h>
58 #include <sys/types.h>
59 #include <sys/stat.h>
60 #include <sys/sockio.h>
61 #include <dirent.h>
62 #include <errno.h>
63 #include <rpccsvc/sm_inter.h>
64 #include <rpccsvc/nsm_addr.h>
65 #include <thread.h>
66 #include <synch.h>
67 #include <net/if.h>
68 #include <limits.h>
69 #include <rpccsvc/daemon_utils.h>
70 #include <priv_utils.h>
71 #include "smfcfg.h"
72 #include "sm_statd.h"

75 #define home0          "/var/statmon"
76 #define current0       "/var/statmon/sm"
77 #define backup0         "/var/statmon/sm.bak"
78 #define state0          "/var/statmon/state"

80 #define home1          "statmon"
81 #define current1       "statmon/sm/"
82 #define backup1         "statmon/sm.bak/"
83 #define state1          "statmon/state"

85 extern int      daemonize_init(void);
86 extern void     daemonize_fini(int fd);

88 /*
89 * User and group IDs to run as. These are hardwired, rather than looked
90 * up at runtime, because they are very unlikely to change and because they
91 * provide some protection against bogus changes to the passwd and group
92 * files.
93 */
94 uid_t   daemon_uid = DAEMON_UID;
95 gid_t   daemon_gid = DAEMON_GID;

97 char STATE[MAXPATHLEN], CURRENT[MAXPATHLEN], BACKUP[MAXPATHLEN];
98 static char statd_home[MAXPATHLEN];

100 int debug;
101 int regfiles_only = 0;           /* 1 => use symlinks in statmon, 0 => don't */
102 int statd_port = 0;
103 char hostname[MAXHOSTNAMELEN];

105 /*
106 * These variables will be used to store all the
107 * alias names for the host, as well as the -a
108 * command line hostnames.
109 */
110 int host_name_count;
111 char **host_name; /* store -a opts */
112 int addrfix; /* # of -a entries */

115 /*
116 * The following 2 variables are meaningful
117 * only under a HA configuration.
118 * The path_name array is dynamically allocated in main() during
119 * command line argument processing for the -p options.
```

2

```

120 */
121 char **path_name = NULL; /* store -p opts */
122 int pathix = 0; /* # of -p entries */
123
124 /* Global variables. Refer to sm_statd.h for description */
125 mutex_t crash_lock;
126 int die;
127 int in_crash;
128 mutex_t sm_trylock;
129 rwlock_t thr_rwlock;
130 cond_t retrywait;
131 mutex_t name_addrlock;
132
133 mutex_t merges_lock;
134 cond_t merges_cond;
135 boolean_t in_merges;
136
137 /* forward references */
138 static void set_statmon_owner(void);
139 static void copy_client_names(void);
140 static void one_statmon_owner(const char *);
141 static int nftw_owner(const char *, const struct stat *, int, struct FTW *);
142
143 /*
144 * statd protocol
145 * commands:
146 *      SM_STAT
147 *          returns stat_fail to caller
148 *      SM_MON
149 *          adds an entry to the monitor_q and the record_q.
150 *          This message is sent by the server lockd to the server
151 *          statd, to indicate that a new client is to be monitored.
152 *          It is also sent by the server lockd to the client statd
153 *          to indicate that a new server is to be monitored.
154 *      SM_UNMON
155 *          removes an entry from the monitor_q and the record_q
156 *      SM_UNMON_ALL
157 *          removes all entries from a particular host from the
158 *          monitor_q and the record_q. Our statd has this
159 *          disabled.
160 *      SM_SIMU_CRASH
161 *          simulate a crash. Removes everything from the
162 *          record_q and the recovery_q, then calls statd_init()
163 *          to restart things. This message is sent by the server
164 *          lockd to the server statd to have all clients notified
165 *          that they should reclaim locks.
166 *      SM_NOTIFY
167 *          Sent by statd on server to statd on client during
168 *          crash recovery. The client statd passes the info
169 *          to its lockd so it can attempt to reclaim the locks
170 *          held on the server.
171
172 * There are three main hash tables used to keep track of things.
173 *      mon_table
174 *          table that keeps track hosts statd must watch. If one of
175 *          these hosts crashes, then any locks held by that host must
176 *          be released.
177 *      record_table
178 *          used to keep track of all the hostname files stored in
179 *          the directory /var/statmon/sm. These are client hosts who
180 *          are holding or have held a lock at some point. Needed
181 *          to determine if a file needs to be created for host in
182 *          /var/statmon/sm.
183 *      recov_q
184 *          used to keep track hostnames during a recovery
185 *

```

```

186 * The entries are hashed based upon the name.
187 *
188 * There is a directory /var/statmon/sm which holds a file named
189 * for each host that is holding (or has held) a lock. This is
190 * used during initialization on startup, or after a simulated
191 * crash.
192 */
193
194 static void
195 sm_prog_1(struct svc_req *rqstp, SVCXPRT *transp)
196 {
197     union {
198         struct sm_name sm_stat_1_arg;
199         struct mon sm_mon_1_arg;
200         struct mon_id sm_unmon_1_arg;
201         struct my_id sm_unmon_all_1_arg;
202         struct stat_chge ntf_arg;
203         struct reglargs regl_arg;
204     } argument;
205
206     union {
207         sm_stat_res stat_resp;
208         sm_stat mon_resp;
209         struct reglres regl_resp;
210     } result;
211
212     bool_t (*xdr_argument)(), (*xdr_result)();
213     char *(*local)();
214
215     /*
216     * Dispatch according to which protocol is being used:
217     *      NSM_ADDR_PROGRAM is the private lockd address
218     *          registration protocol.
219     *      SM_PROG is the normal statd (NSM) protocol.
220     */
221     if (rqstp->rq_prog == NSM_ADDR_PROGRAM) {
222         switch (rqstp->rq_proc) {
223             case NULLPROC:
224                 svc_sendreply(transp, xdr_void, (caddr_t)NULL);
225                 return;
226
227             case NSMADDRPROC1_REG:
228                 xdr_argument = xdr_reglargs;
229                 xdr_result = xdr_reglres;
230                 local = (char *(*())() nsmaddrproc1_reg;
231                 break;
232
233             case NSMADDRPROC1_UNREG: /* Not impl. */
234             default:
235                 svcerr_noproc(transp);
236                 return;
237         }
238     } else {
239         /* Must be SM_PROG */
240         switch (rqstp->rq_proc) {
241             case NULLPROC:
242                 svc_sendreply(transp, xdr_void, (caddr_t)NULL);
243                 return;
244
245             case SM_STAT:
246                 xdr_argument = xdr_sm_name;
247                 xdr_result = xdr_sm_stat_res;
248                 local = (char *(*())() sm_stat_svc;
249                 break;
250
251             case SM_MON:
252
253         }
254     }
255
256     if (argument.sm_stat_1_arg.name != NULL)
257         sm_stat(transp, &argument.sm_stat_1_arg, &result.stat_resp);
258     else if (argument.sm_mon_1_arg.host != NULL)
259         sm_mon(transp, &argument.sm_mon_1_arg, &result.mon_resp);
260     else if (argument.sm_unmon_1_arg.host != NULL)
261         sm_unmon(transp, &argument.sm_unmon_1_arg, &result.mon_resp);
262     else if (argument.sm_unmon_all_1_arg.host != NULL)
263         sm_unmon_all(transp, &argument.sm_unmon_all_1_arg, &result.mon_resp);
264     else if (argument.ntf_arg.host != NULL)
265         sm_ntf(transp, &argument.ntf_arg, &result.stat_resp);
266     else if (argument.regl_arg.host != NULL)
267         sm_regl(transp, &argument.regl_arg, &result.stat_resp);
268
269     if (local != NULL)
270         local();
271
272     if (xdr_result())
273         svc_sendreply(transp, xdr_void, (caddr_t)local);
274     else
275         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
276
277     if (local != NULL)
278         local();
279
280     if (rqstp->rq_error != 0)
281         svc_senderror(transp, rqstp->rq_error);
282
283     if (rqstp->rq_error == 0)
284         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
285
286     if (rqstp->rq_error != 0)
287         svc_senderror(transp, rqstp->rq_error);
288
289     if (rqstp->rq_error == 0)
290         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
291
292     if (rqstp->rq_error != 0)
293         svc_senderror(transp, rqstp->rq_error);
294
295     if (rqstp->rq_error == 0)
296         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
297
298     if (rqstp->rq_error != 0)
299         svc_senderror(transp, rqstp->rq_error);
300
301     if (rqstp->rq_error == 0)
302         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
303
304     if (rqstp->rq_error != 0)
305         svc_senderror(transp, rqstp->rq_error);
306
307     if (rqstp->rq_error == 0)
308         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
309
310     if (rqstp->rq_error != 0)
311         svc_senderror(transp, rqstp->rq_error);
312
313     if (rqstp->rq_error == 0)
314         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
315
316     if (rqstp->rq_error != 0)
317         svc_senderror(transp, rqstp->rq_error);
318
319     if (rqstp->rq_error == 0)
320         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
321
322     if (rqstp->rq_error != 0)
323         svc_senderror(transp, rqstp->rq_error);
324
325     if (rqstp->rq_error == 0)
326         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
327
328     if (rqstp->rq_error != 0)
329         svc_senderror(transp, rqstp->rq_error);
330
331     if (rqstp->rq_error == 0)
332         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
333
334     if (rqstp->rq_error != 0)
335         svc_senderror(transp, rqstp->rq_error);
336
337     if (rqstp->rq_error == 0)
338         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
339
340     if (rqstp->rq_error != 0)
341         svc_senderror(transp, rqstp->rq_error);
342
343     if (rqstp->rq_error == 0)
344         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
345
346     if (rqstp->rq_error != 0)
347         svc_senderror(transp, rqstp->rq_error);
348
349     if (rqstp->rq_error == 0)
350         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
351
352     if (rqstp->rq_error != 0)
353         svc_senderror(transp, rqstp->rq_error);
354
355     if (rqstp->rq_error == 0)
356         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
357
358     if (rqstp->rq_error != 0)
359         svc_senderror(transp, rqstp->rq_error);
360
361     if (rqstp->rq_error == 0)
362         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
363
364     if (rqstp->rq_error != 0)
365         svc_senderror(transp, rqstp->rq_error);
366
367     if (rqstp->rq_error == 0)
368         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
369
370     if (rqstp->rq_error != 0)
371         svc_senderror(transp, rqstp->rq_error);
372
373     if (rqstp->rq_error == 0)
374         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
375
376     if (rqstp->rq_error != 0)
377         svc_senderror(transp, rqstp->rq_error);
378
379     if (rqstp->rq_error == 0)
380         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
381
382     if (rqstp->rq_error != 0)
383         svc_senderror(transp, rqstp->rq_error);
384
385     if (rqstp->rq_error == 0)
386         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
387
388     if (rqstp->rq_error != 0)
389         svc_senderror(transp, rqstp->rq_error);
390
391     if (rqstp->rq_error == 0)
392         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
393
394     if (rqstp->rq_error != 0)
395         svc_senderror(transp, rqstp->rq_error);
396
397     if (rqstp->rq_error == 0)
398         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
399
400     if (rqstp->rq_error != 0)
401         svc_senderror(transp, rqstp->rq_error);
402
403     if (rqstp->rq_error == 0)
404         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
405
406     if (rqstp->rq_error != 0)
407         svc_senderror(transp, rqstp->rq_error);
408
409     if (rqstp->rq_error == 0)
410         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
411
412     if (rqstp->rq_error != 0)
413         svc_senderror(transp, rqstp->rq_error);
414
415     if (rqstp->rq_error == 0)
416         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
417
418     if (rqstp->rq_error != 0)
419         svc_senderror(transp, rqstp->rq_error);
420
421     if (rqstp->rq_error == 0)
422         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
423
424     if (rqstp->rq_error != 0)
425         svc_senderror(transp, rqstp->rq_error);
426
427     if (rqstp->rq_error == 0)
428         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
429
430     if (rqstp->rq_error != 0)
431         svc_senderror(transp, rqstp->rq_error);
432
433     if (rqstp->rq_error == 0)
434         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
435
436     if (rqstp->rq_error != 0)
437         svc_senderror(transp, rqstp->rq_error);
438
439     if (rqstp->rq_error == 0)
440         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
441
442     if (rqstp->rq_error != 0)
443         svc_senderror(transp, rqstp->rq_error);
444
445     if (rqstp->rq_error == 0)
446         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
447
448     if (rqstp->rq_error != 0)
449         svc_senderror(transp, rqstp->rq_error);
450
451     if (rqstp->rq_error == 0)
452         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
453
454     if (rqstp->rq_error != 0)
455         svc_senderror(transp, rqstp->rq_error);
456
457     if (rqstp->rq_error == 0)
458         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
459
460     if (rqstp->rq_error != 0)
461         svc_senderror(transp, rqstp->rq_error);
462
463     if (rqstp->rq_error == 0)
464         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
465
466     if (rqstp->rq_error != 0)
467         svc_senderror(transp, rqstp->rq_error);
468
469     if (rqstp->rq_error == 0)
470         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
471
472     if (rqstp->rq_error != 0)
473         svc_senderror(transp, rqstp->rq_error);
474
475     if (rqstp->rq_error == 0)
476         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
477
478     if (rqstp->rq_error != 0)
479         svc_senderror(transp, rqstp->rq_error);
480
481     if (rqstp->rq_error == 0)
482         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
483
484     if (rqstp->rq_error != 0)
485         svc_senderror(transp, rqstp->rq_error);
486
487     if (rqstp->rq_error == 0)
488         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
489
490     if (rqstp->rq_error != 0)
491         svc_senderror(transp, rqstp->rq_error);
492
493     if (rqstp->rq_error == 0)
494         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
495
496     if (rqstp->rq_error != 0)
497         svc_senderror(transp, rqstp->rq_error);
498
499     if (rqstp->rq_error == 0)
500         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
501
502     if (rqstp->rq_error != 0)
503         svc_senderror(transp, rqstp->rq_error);
504
505     if (rqstp->rq_error == 0)
506         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
507
508     if (rqstp->rq_error != 0)
509         svc_senderror(transp, rqstp->rq_error);
510
511     if (rqstp->rq_error == 0)
512         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
513
514     if (rqstp->rq_error != 0)
515         svc_senderror(transp, rqstp->rq_error);
516
517     if (rqstp->rq_error == 0)
518         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
519
520     if (rqstp->rq_error != 0)
521         svc_senderror(transp, rqstp->rq_error);
522
523     if (rqstp->rq_error == 0)
524         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
525
526     if (rqstp->rq_error != 0)
527         svc_senderror(transp, rqstp->rq_error);
528
529     if (rqstp->rq_error == 0)
530         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
531
532     if (rqstp->rq_error != 0)
533         svc_senderror(transp, rqstp->rq_error);
534
535     if (rqstp->rq_error == 0)
536         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
537
538     if (rqstp->rq_error != 0)
539         svc_senderror(transp, rqstp->rq_error);
540
541     if (rqstp->rq_error == 0)
542         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
543
544     if (rqstp->rq_error != 0)
545         svc_senderror(transp, rqstp->rq_error);
546
547     if (rqstp->rq_error == 0)
548         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
549
550     if (rqstp->rq_error != 0)
551         svc_senderror(transp, rqstp->rq_error);
552
553     if (rqstp->rq_error == 0)
554         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
555
556     if (rqstp->rq_error != 0)
557         svc_senderror(transp, rqstp->rq_error);
558
559     if (rqstp->rq_error == 0)
560         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
561
562     if (rqstp->rq_error != 0)
563         svc_senderror(transp, rqstp->rq_error);
564
565     if (rqstp->rq_error == 0)
566         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
567
568     if (rqstp->rq_error != 0)
569         svc_senderror(transp, rqstp->rq_error);
570
571     if (rqstp->rq_error == 0)
572         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
573
574     if (rqstp->rq_error != 0)
575         svc_senderror(transp, rqstp->rq_error);
576
577     if (rqstp->rq_error == 0)
578         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
579
580     if (rqstp->rq_error != 0)
581         svc_senderror(transp, rqstp->rq_error);
582
583     if (rqstp->rq_error == 0)
584         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
585
586     if (rqstp->rq_error != 0)
587         svc_senderror(transp, rqstp->rq_error);
588
589     if (rqstp->rq_error == 0)
590         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
591
592     if (rqstp->rq_error != 0)
593         svc_senderror(transp, rqstp->rq_error);
594
595     if (rqstp->rq_error == 0)
596         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
597
598     if (rqstp->rq_error != 0)
599         svc_senderror(transp, rqstp->rq_error);
599
600     if (rqstp->rq_error == 0)
601         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
601
602     if (rqstp->rq_error != 0)
603         svc_senderror(transp, rqstp->rq_error);
603
604     if (rqstp->rq_error == 0)
605         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
605
606     if (rqstp->rq_error != 0)
607         svc_senderror(transp, rqstp->rq_error);
607
608     if (rqstp->rq_error == 0)
609         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
609
610     if (rqstp->rq_error != 0)
611         svc_senderror(transp, rqstp->rq_error);
611
612     if (rqstp->rq_error == 0)
613         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
613
614     if (rqstp->rq_error != 0)
615         svc_senderror(transp, rqstp->rq_error);
615
616     if (rqstp->rq_error == 0)
617         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
617
618     if (rqstp->rq_error != 0)
619         svc_senderror(transp, rqstp->rq_error);
619
620     if (rqstp->rq_error == 0)
621         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
621
622     if (rqstp->rq_error != 0)
623         svc_senderror(transp, rqstp->rq_error);
623
624     if (rqstp->rq_error == 0)
625         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
625
626     if (rqstp->rq_error != 0)
627         svc_senderror(transp, rqstp->rq_error);
627
628     if (rqstp->rq_error == 0)
629         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
629
630     if (rqstp->rq_error != 0)
631         svc_senderror(transp, rqstp->rq_error);
631
632     if (rqstp->rq_error == 0)
633         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
633
634     if (rqstp->rq_error != 0)
635         svc_senderror(transp, rqstp->rq_error);
635
636     if (rqstp->rq_error == 0)
637         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
637
638     if (rqstp->rq_error != 0)
639         svc_senderror(transp, rqstp->rq_error);
639
640     if (rqstp->rq_error == 0)
641         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
641
642     if (rqstp->rq_error != 0)
643         svc_senderror(transp, rqstp->rq_error);
643
644     if (rqstp->rq_error == 0)
645         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
645
646     if (rqstp->rq_error != 0)
647         svc_senderror(transp, rqstp->rq_error);
647
648     if (rqstp->rq_error == 0)
649         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
649
650     if (rqstp->rq_error != 0)
651         svc_senderror(transp, rqstp->rq_error);
651
652     if (rqstp->rq_error == 0)
653         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
653
654     if (rqstp->rq_error != 0)
655         svc_senderror(transp, rqstp->rq_error);
655
656     if (rqstp->rq_error == 0)
657         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
657
658     if (rqstp->rq_error != 0)
659         svc_senderror(transp, rqstp->rq_error);
659
660     if (rqstp->rq_error == 0)
661         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
661
662     if (rqstp->rq_error != 0)
663         svc_senderror(transp, rqstp->rq_error);
663
664     if (rqstp->rq_error == 0)
665         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
665
666     if (rqstp->rq_error != 0)
667         svc_senderror(transp, rqstp->rq_error);
667
668     if (rqstp->rq_error == 0)
669         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
669
670     if (rqstp->rq_error != 0)
671         svc_senderror(transp, rqstp->rq_error);
671
672     if (rqstp->rq_error == 0)
673         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
673
674     if (rqstp->rq_error != 0)
675         svc_senderror(transp, rqstp->rq_error);
675
676     if (rqstp->rq_error == 0)
677         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
677
678     if (rqstp->rq_error != 0)
679         svc_senderror(transp, rqstp->rq_error);
679
680     if (rqstp->rq_error == 0)
681         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
681
682     if (rqstp->rq_error != 0)
683         svc_senderror(transp, rqstp->rq_error);
683
684     if (rqstp->rq_error == 0)
685         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
685
686     if (rqstp->rq_error != 0)
687         svc_senderror(transp, rqstp->rq_error);
687
688     if (rqstp->rq_error == 0)
689         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
689
690     if (rqstp->rq_error != 0)
691         svc_senderror(transp, rqstp->rq_error);
691
692     if (rqstp->rq_error == 0)
693         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
693
694     if (rqstp->rq_error != 0)
695         svc_senderror(transp, rqstp->rq_error);
695
696     if (rqstp->rq_error == 0)
697         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
697
698     if (rqstp->rq_error != 0)
699         svc_senderror(transp, rqstp->rq_error);
699
700     if (rqstp->rq_error == 0)
701         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
701
702     if (rqstp->rq_error != 0)
703         svc_senderror(transp, rqstp->rq_error);
703
704     if (rqstp->rq_error == 0)
705         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
705
706     if (rqstp->rq_error != 0)
707         svc_senderror(transp, rqstp->rq_error);
707
708     if (rqstp->rq_error == 0)
709         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
709
710     if (rqstp->rq_error != 0)
711         svc_senderror(transp, rqstp->rq_error);
711
712     if (rqstp->rq_error == 0)
713         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
713
714     if (rqstp->rq_error != 0)
715         svc_senderror(transp, rqstp->rq_error);
715
716     if (rqstp->rq_error == 0)
717         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
717
718     if (rqstp->rq_error != 0)
719         svc_senderror(transp, rqstp->rq_error);
719
720     if (rqstp->rq_error == 0)
721         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
721
722     if (rqstp->rq_error != 0)
723         svc_senderror(transp, rqstp->rq_error);
723
724     if (rqstp->rq_error == 0)
725         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
725
726     if (rqstp->rq_error != 0)
727         svc_senderror(transp, rqstp->rq_error);
727
728     if (rqstp->rq_error == 0)
729         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
729
730     if (rqstp->rq_error != 0)
731         svc_senderror(transp, rqstp->rq_error);
731
732     if (rqstp->rq_error == 0)
733         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
733
734     if (rqstp->rq_error != 0)
735         svc_senderror(transp, rqstp->rq_error);
735
736     if (rqstp->rq_error == 0)
737         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
737
738     if (rqstp->rq_error != 0)
739         svc_senderror(transp, rqstp->rq_error);
739
740     if (rqstp->rq_error == 0)
741         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
741
742     if (rqstp->rq_error != 0)
743         svc_senderror(transp, rqstp->rq_error);
743
744     if (rqstp->rq_error == 0)
745         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
745
746     if (rqstp->rq_error != 0)
747         svc_senderror(transp, rqstp->rq_error);
747
748     if (rqstp->rq_error == 0)
749         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
749
750     if (rqstp->rq_error != 0)
751         svc_senderror(transp, rqstp->rq_error);
751
752     if (rqstp->rq_error == 0)
753         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
753
754     if (rqstp->rq_error != 0)
755         svc_senderror(transp, rqstp->rq_error);
755
756     if (rqstp->rq_error == 0)
757         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
757
758     if (rqstp->rq_error != 0)
759         svc_senderror(transp, rqstp->rq_error);
759
760     if (rqstp->rq_error == 0)
761         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
761
762     if (rqstp->rq_error != 0)
763         svc_senderror(transp, rqstp->rq_error);
763
764     if (rqstp->rq_error == 0)
765         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
765
766     if (rqstp->rq_error != 0)
767         svc_senderror(transp, rqstp->rq_error);
767
768     if (rqstp->rq_error == 0)
769         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
769
770     if (rqstp->rq_error != 0)
771         svc_senderror(transp, rqstp->rq_error);
771
772     if (rqstp->rq_error == 0)
773         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
773
774     if (rqstp->rq_error != 0)
775         svc_senderror(transp, rqstp->rq_error);
775
776     if (rqstp->rq_error == 0)
777         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
777
778     if (rqstp->rq_error != 0)
779         svc_senderror(transp, rqstp->rq_error);
779
780     if (rqstp->rq_error == 0)
781         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
781
782     if (rqstp->rq_error != 0)
783         svc_senderror(transp, rqstp->rq_error);
783
784     if (rqstp->rq_error == 0)
785         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
785
786     if (rqstp->rq_error != 0)
787         svc_senderror(transp, rqstp->rq_error);
787
788     if (rqstp->rq_error == 0)
789         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
789
790     if (rqstp->rq_error != 0)
791         svc_senderror(transp, rqstp->rq_error);
791
792     if (rqstp->rq_error == 0)
793         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
793
794     if (rqstp->rq_error != 0)
795         svc_senderror(transp, rqstp->rq_error);
795
796     if (rqstp->rq_error == 0)
797         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
797
798     if (rqstp->rq_error != 0)
799         svc_senderror(transp, rqstp->rq_error);
799
800     if (rqstp->rq_error == 0)
801         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
801
802     if (rqstp->rq_error != 0)
803         svc_senderror(transp, rqstp->rq_error);
803
804     if (rqstp->rq_error == 0)
805         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
805
806     if (rqstp->rq_error != 0)
807         svc_senderror(transp, rqstp->rq_error);
807
808     if (rqstp->rq_error == 0)
809         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
809
810     if (rqstp->rq_error != 0)
811         svc_senderror(transp, rqstp->rq_error);
811
812     if (rqstp->rq_error == 0)
813         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
813
814     if (rqstp->rq_error != 0)
815         svc_senderror(transp, rqstp->rq_error);
815
816     if (rqstp->rq_error == 0)
817         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
817
818     if (rqstp->rq_error != 0)
819         svc_senderror(transp, rqstp->rq_error);
819
820     if (rqstp->rq_error == 0)
821         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
821
822     if (rqstp->rq_error != 0)
823         svc_senderror(transp, rqstp->rq_error);
823
824     if (rqstp->rq_error == 0)
825         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
825
826     if (rqstp->rq_error != 0)
827         svc_senderror(transp, rqstp->rq_error);
827
828     if (rqstp->rq_error == 0)
829         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
829
830     if (rqstp->rq_error != 0)
831         svc_senderror(transp, rqstp->rq_error);
831
832     if (rqstp->rq_error == 0)
833         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
833
834     if (rqstp->rq_error != 0)
835         svc_senderror(transp, rqstp->rq_error);
835
836     if (rqstp->rq_error == 0)
837         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
837
838     if (rqstp->rq_error != 0)
839         svc_senderror(transp, rqstp->rq_error);
839
840     if (rqstp->rq_error == 0)
841         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
841
842     if (rqstp->rq_error != 0)
843         svc_senderror(transp, rqstp->rq_error);
843
844     if (rqstp->rq_error == 0)
845         svc_sendreply(transp, xdr_void, (caddr_t)NULL);
845
846     if (rqstp->rq_error != 0
```

```

252         xdr_argument = xdr_mon;
253         xdr_result = xdr_sm_stat_res;
254         local = (char *(*)()) sm_mon_svc;
255         break;
256
257     case SM_UNMON:
258         xdr_argument = xdr_mon_id;
259         xdr_result = xdr_sm_stat;
260         local = (char *(*)()) sm_unmon_svc;
261         break;
262
263     case SM_UNMON_ALL:
264         xdr_argument = xdr_my_id;
265         xdr_result = xdr_sm_stat;
266         local = (char *(*)()) sm_unmon_all_svc;
267         break;
268
269     case SM_SIMU_CRASH:
270         xdr_argument = xdr_void;
271         xdr_result = xdr_void;
272         local = (char *(*)()) sm_simu_crash_svc;
273         break;
274
275     case SM_NOTIFY:
276         xdr_argument = xdr_stat_chge;
277         xdr_result = xdr_void;
278         local = (char *(*)()) sm_notify_svc;
279         break;
280
281     default:
282         svcerr_noproc(transp);
283         return;
284     }
285 }
286
287 (void) memset(&argument, 0, sizeof (argument));
288 if (!svc_getargs(transp, xdr_argument, (caddr_t)&argument)) {
289     svcerr_decode(transp);
290     return;
291 }
292
293 (void) memset(&result, 0, sizeof (result));
294 (*local)(&argument, &result);
295 if (!svc_sendreply(transp, xdr_result, (caddr_t)&result)) {
296     svcerr_systemerr(transp);
297 }
298
299 if (!svc_freeargs(transp, xdr_argument, (caddr_t)&argument)) {
300     syslog(LOG_ERR, "statd: unable to free arguments\n");
301 }
302 }
303
304 unchanged_portion_omitted
305
306 /*
307  * This function is called for each configured network type to
308  * bind and register our RPC service programs.
309  *
310  * On TCP or UDP, we may want to bind SM_PROG on a specific port
311  * (when statd_port is specified) in which case we'll use the
312  * variant of svc_tp_create() that lets us pass a bind address.
313  */
314 static void
315 sm_svc_tp_create(struct netconfig *nconf)
316 {
317     char port_str[8];
318     struct nd_hostserv hs;
319
320     /* ... omitted code ...
321      */
322
323     /* ... omitted code ...
324      */
325
326     /* ... omitted code ...
327      */
328
329     /* ... omitted code ...
330      */
331
332     /* ... omitted code ...
333      */
334
335     /* ... omitted code ...
336      */
337
338     /* ... omitted code ...
339      */
340
341     /* ... omitted code ...
342      */
343
344     /* ... omitted code ...
345      */
346
347     /* ... omitted code ...
348      */
349
350     /* ... omitted code ...
351      */
352
353     /* ... omitted code ...
354      */
355
356     /* ... omitted code ...
357      */
358
359     /* ... omitted code ...
360      */
361
362     /* ... omitted code ...
363      */
364
365     /* ... omitted code ...
366      */
367
368     /* ... omitted code ...
369      */
370
371     /* ... omitted code ...
372      */
373
374     /* ... omitted code ...
375      */
376
377     /* ... omitted code ...
378      */
379
380     /* ... omitted code ...
381      */
382
383     /* ... omitted code ...
384      */
385
386     /* ... omitted code ...
387      */
388
389     /* ... omitted code ...
390      */
391
392     /* ... omitted code ...
393      */
394
395     /* ... omitted code ...
396      */
397
398     /* ... omitted code ...
399      */
400
401     /* ... omitted code ...
402      */
403
404     /* ... omitted code ...
405      */
406
407     /* ... omitted code ...
408      */
409
410     /* ... omitted code ...
411      */
412
413     /* ... omitted code ...
414      */
415
416     /* ... omitted code ...
417      */
418
419     /* ... omitted code ...
420      */
421
422     /* ... omitted code ...
423      */
424
425     /* ... omitted code ...
426      */
427
428     /* ... omitted code ...
429      */
430
431     /* ... omitted code ...
432      */
433
434     /* ... omitted code ...
435      */
436
437     /* ... omitted code ...
438      */
439
440     /* ... omitted code ...
441      */
442
443     /* ... omitted code ...
444      */
445
446     /* ... omitted code ...
447      */
448
449     /* ... omitted code ...
450      */
451
452     /* ... omitted code ...
453      */
454
455     /* ... omitted code ...
456      */
457
458     /* ... omitted code ...
459      */
460
461     /* ... omitted code ...
462      */
463
464     /* ... omitted code ...
465      */
466
467     /* ... omitted code ...
468      */
469
470     /* ... omitted code ...
471      */
472
473     /* ... omitted code ...
474      */
475
476     /* ... omitted code ...
477      */
478
479     /* ... omitted code ...
480      */
481
482     /* ... omitted code ...
483      */
484
485     /* ... omitted code ...
486      */
487
488     /* ... omitted code ...
489      */
490
491     /* ... omitted code ...
492      */
493
494     /* ... omitted code ...
495      */
496
497     /* ... omitted code ...
498      */
499
500     /* ... omitted code ...
501      */
502
503     /* ... omitted code ...
504      */
505
506     /* ... omitted code ...
507      */
508
509     /* ... omitted code ...
510      */
511
512     /* ... omitted code ...
513      */
514
515     /* ... omitted code ...
516      */
517
518     /* ... omitted code ...
519      */
520
521     /* ... omitted code ...
522      */
523
524     /* ... omitted code ...
525      */
526
527     /* ... omitted code ...
528      */
529
530     /* ... omitted code ...
531      */
532
533     /* ... omitted code ...
534      */
535
536     /* ... omitted code ...
537      */
538
539     /* ... omitted code ...
540      */
541
542     /* ... omitted code ...
543      */
544
545     /* ... omitted code ...
546      */
547
548     /* ... omitted code ...
549      */
550
551     /* ... omitted code ...
552      */
553
554     /* ... omitted code ...
555      */
556
557     /* ... omitted code ...
558      */
559
560     /* ... omitted code ...
561      */
562
563     /* ... omitted code ...
564      */
565
566     /* ... omitted code ...
567      */
568
569     /* ... omitted code ...
570      */
571
572     /* ... omitted code ...
573      */
574
575     /* ... omitted code ...
576      */
577
578     /* ... omitted code ...
579      */
580
581     /* ... omitted code ...
582      */
583
584     /* ... omitted code ...
585      */
586
587     /* ... omitted code ...
588      */
589
590     /* ... omitted code ...
591      */
592
593     /* ... omitted code ...
594      */
595
596     /* ... omitted code ...
597      */
598
599     /* ... omitted code ...
600      */
601
602     /* ... omitted code ...
603      */
604
605     /* ... omitted code ...
606      */
607
608     /* ... omitted code ...
609      */
610
611     /* ... omitted code ...
612      */
613
614     /* ... omitted code ...
615      */
616
617     /* ... omitted code ...
618      */
619
620     /* ... omitted code ...
621      */
622
623     /* ... omitted code ...
624      */
625
626     /* ... omitted code ...
627      */
628
629     /* ... omitted code ...
630      */
631
632     /* ... omitted code ...
633      */
634
635     /* ... omitted code ...
636      */
637
638     /* ... omitted code ...
639      */
640
641     /* ... omitted code ...
642      */
643
644     /* ... omitted code ...
645      */
646
647     /* ... omitted code ...
648      */
649
650     /* ... omitted code ...
651      */
652
653     /* ... omitted code ...
654      */
655
656     /* ... omitted code ...
657      */
658
659     /* ... omitted code ...
660      */
661
662     /* ... omitted code ...
663      */
664
665     /* ... omitted code ...
666      */
667
668     /* ... omitted code ...
669      */
670
671     /* ... omitted code ...
672      */
673
674     /* ... omitted code ...
675      */
676
677     /* ... omitted code ...
678      */
679
680     /* ... omitted code ...
681      */
682
683     /* ... omitted code ...
684      */
685
686     /* ... omitted code ...
687      */
688
689     /* ... omitted code ...
690      */
691
692     /* ... omitted code ...
693      */
694
695     /* ... omitted code ...
696      */
697
698     /* ... omitted code ...
699      */
700
701     /* ... omitted code ...
702      */
703
704     /* ... omitted code ...
705      */
706
707     /* ... omitted code ...
708      */
709
710     /* ... omitted code ...
711      */
712
713     /* ... omitted code ...
714      */
715
716     /* ... omitted code ...
717      */
718
719     /* ... omitted code ...
720      */
721
722     /* ... omitted code ...
723      */
724
725     /* ... omitted code ...
726      */
727
728     /* ... omitted code ...
729      */
730
731     /* ... omitted code ...
732      */
733
734     /* ... omitted code ...
735      */
736
737     /* ... omitted code ...
738      */
739
740     /* ... omitted code ...
741      */
742
743     /* ... omitted code ...
744      */
745
746     /* ... omitted code ...
747      */
748
749     /* ... omitted code ...
750      */
751
752     /* ... omitted code ...
753      */
754
755     /* ... omitted code ...
756      */
757
758     /* ... omitted code ...
759      */
760
761     /* ... omitted code ...
762      */
763
764     /* ... omitted code ...
765      */
766
767     /* ... omitted code ...
768      */
769
770     /* ... omitted code ...
771      */
772
773     /* ... omitted code ...
774      */
775
776     /* ... omitted code ...
777      */
778
779     /* ... omitted code ...
780      */
781
782     /* ... omitted code ...
783      */
784
785     /* ... omitted code ...
786      */
787
788     /* ... omitted code ...
789      */
790
791     /* ... omitted code ...
792      */
793
794     /* ... omitted code ...
795      */
796
797     /* ... omitted code ...
798      */
799
800     /* ... omitted code ...
801      */
802
803     /* ... omitted code ...
804      */
805
806     /* ... omitted code ...
807      */
808
809     /* ... omitted code ...
810      */
811
812     /* ... omitted code ...
813      */
814
815     /* ... omitted code ...
816      */
817
818     /* ... omitted code ...
819      */
820
821     /* ... omitted code ...
822      */
823
824     /* ... omitted code ...
825      */
826
827     /* ... omitted code ...
828      */
829
830     /* ... omitted code ...
831      */
832
833     /* ... omitted code ...
834      */
835
836     /* ... omitted code ...
837      */
838
839     /* ... omitted code ...
840      */
841
842     /* ... omitted code ...
843      */
844
845     /* ... omitted code ...
846      */
847
848     /* ... omitted code ...
849      */
850
851     /* ... omitted code ...
852      */
853
854     /* ... omitted code ...
855      */
856
857     /* ... omitted code ...
858      */
859
860     /* ... omitted code ...
861      */
862
863     /* ... omitted code ...
864      */
865
866     /* ... omitted code ...
867      */
868
869     /* ... omitted code ...
870      */
871
872     /* ... omitted code ...
873      */
874
875     /* ... omitted code ...
876      */
877
878     /* ... omitted code ...
879      */
880
881     /* ... omitted code ...
882      */
883
884     /* ... omitted code ...
885      */
886
887     /* ... omitted code ...
888      */
889
890     /* ... omitted code ...
891      */
892
893     /* ... omitted code ...
894      */
895
896     /* ... omitted code ...
897      */
898
899     /* ... omitted code ...
900      */
901
902     /* ... omitted code ...
903      */
904
905     /* ... omitted code ...
906      */
907
908     /* ... omitted code ...
909      */
910
911     /* ... omitted code ...
912      */
913
914     /* ... omitted code ...
915      */
916
917     /* ... omitted code ...
918      */
919
920     /* ... omitted code ...
921      */
922
923     /* ... omitted code ...
924      */
925
926     /* ... omitted code ...
927      */
928
929     /* ... omitted code ...
930      */
931
932     /* ... omitted code ...
933      */
934
935     /* ... omitted code ...
936      */
937
938     /* ... omitted code ...
939      */
940
941     /* ... omitted code ...
942      */
943
944     /* ... omitted code ...
945      */
946
947     /* ... omitted code ...
948      */
949
950     /* ... omitted code ...
951      */
952
953     /* ... omitted code ...
954      */
955
956     /* ... omitted code ...
957      */
958
959     /* ... omitted code ...
960      */
961
962     /* ... omitted code ...
963      */
964
965     /* ... omitted code ...
966      */
967
968     /* ... omitted code ...
969      */
970
971     /* ... omitted code ...
972      */
973
974     /* ... omitted code ...
975      */
976
977     /* ... omitted code ...
978      */
979
980     /* ... omitted code ...
981      */
982
983     /* ... omitted code ...
984      */
985
986     /* ... omitted code ...
987      */
988
989     /* ... omitted code ...
990      */
991
992     /* ... omitted code ...
993      */
994
995     /* ... omitted code ...
996      */
997
998     /* ... omitted code ...
999      */
1000
1001     /* ... omitted code ...
1002      */
1003
1004     /* ... omitted code ...
1005      */
1006
1007     /* ... omitted code ...
1008      */
1009
1010     /* ... omitted code ...
1011      */
1012
1013     /* ... omitted code ...
1014      */
1015
1016     /* ... omitted code ...
1017      */
1018
1019     /* ... omitted code ...
1020      */
1021
1022     /* ... omitted code ...
1023      */
1024
1025     /* ... omitted code ...
1026      */
1027
1028     /* ... omitted code ...
1029      */
1030
1031     /* ... omitted code ...
1032      */
1033
1034     /* ... omitted code ...
1035      */
1036
1037     /* ... omitted code ...
1038      */
1039
1040     /* ... omitted code ...
1041      */
1042
1043     /* ... omitted code ...
1044      */
1045
1046     /* ... omitted code ...
1047      */
1048
1049     /* ... omitted code ...
1050      */
1051
1052     /* ... omitted code ...
1053      */
1054
1055     /* ... omitted code ...
1056      */
1057
1058     /* ... omitted code ...
1059      */
1060
1061     /* ... omitted code ...
1062      */
1063
1064     /* ... omitted code ...
1065      */
1066
1067     /* ... omitted code ...
1068      */
1069
1070     /* ... omitted code ...
1071      */
1072
1073     /* ... omitted code ...
1074      */
1075
1076     /* ... omitted code ...
1077      */
1078
1079     /* ... omitted code ...
1080      */
1081
1082     /* ... omitted code ...
1083      */
1084
1085     /* ... omitted code ...
1086      */
1087
1088     /* ... omitted code ...
1089      */
1090
1091     /* ... omitted code ...
1092      */
1093
1094     /* ... omitted code ...
1095      */
1096
1097     /* ... omitted code ...
1098      */
1099
1100     /* ... omitted code ...
1101      */
1102
1103     /* ... omitted code ...
1104      */
1105
1106     /* ... omitted code ...
1107      */
1108
1109     /* ... omitted code ...
1110      */
1111
1112     /* ... omitted code ...
1113      */
1114
1115     /* ... omitted code ...
1116      */
1117
1118     /* ... omitted code ...
1119      */
1120
1121     /* ... omitted code ...
1122      */
1123
1124     /* ... omitted code ...
1125      */
1126
1127     /* ... omitted code ...
1128      */
1129
1130     /* ... omitted code ...
1131      */
1132
1133     /* ... omitted code ...
1134      */
1135
1136     /* ... omitted code ...
1137      */
1138
1139     /* ... omitted code ...
1140      */
1141
1142     /* ... omitted code ...
1143      */
1144
1145     /* ... omitted code ...
1146      */
1147
1148     /* ... omitted code ...
1149      */
1150
1151     /* ... omitted code ...
1152      */
1153
1154     /* ... omitted code ...
1155      */
1156
1157     /* ... omitted code ...
1158      */
1159
1160     /* ... omitted code ...
1161      */
1162
1163     /* ... omitted code ...
1164      */
1165
1166     /* ... omitted code ...
1167      */
1168
1169     /* ... omitted code ...
1170      */
1171
1172     /* ... omitted code ...
1173      */
1174
1175     /* ... omitted code ...
1176      */
1177
1178     /* ... omitted code ...
1179      */
1180
1181     /* ... omitted code ...
1182      */
1183
1184     /* ... omitted code ...
1185      */
1186
1187     /* ... omitted code ...
1188      */
1189
1190     /* ... omitted code ...
1191      */
1192
1193     /* ... omitted code ...
1194      */
1195
1196     /* ... omitted code ...
1197      */
1198
1199     /* ... omitted code ...
1200      */
1201
1202     /* ... omitted code ...
1203      */
1204
1205     /* ... omitted code ...
1206      */
1207
1208     /* ... omitted code ...
1209      */
1210
1211     /* ... omitted code ...
1212      */
1213
1214     /* ... omitted code ...
1215      */
1216
1217     /* ... omitted code ...
1218      */
1219
1220     /* ... omitted code ...
1221      */
1222
1223     /* ... omitted code ...
1224      */
1225
1226     /* ... omitted code ...
1227      */
1228
1229     /* ... omitted code ...
1230      */
1231
1232     /* ... omitted code ...
1233      */
1234
1235     /* ... omitted code ...
1236      */
1237
1238     /* ... omitted code ...
1239      */
1240
1241     /* ... omitted code ...
1242      */
1243
1244     /* ... omitted code ...
1245      */
1246
1247     /* ... omitted code ...
1248      */
1249
1250     /* ... omitted code ...
1251      */
1252
1253     /* ... omitted code ...
1254      */
1255
1256     /* ... omitted code ...
1257      */
1258
1259     /* ... omitted code ...
1260      */
1261
1262     /* ... omitted code ...
1263      */
1264
1265     /* ... omitted code ...
1266      */
1267
1268     /* ... omitted code ...
1269      */
1270
1271     /* ... omitted code ...
1272      */
1273
1274     /* ... omitted code ...
1275      */
1276
1277     /* ... omitted code ...
1278      */
1279
1280     /* ... omitted code ...
1281      */
1282
1283     /* ... omitted code ...
1284      */
1285
1286     /* ... omitted code ...
1287      */
1288
1289     /* ... omitted code ...
1290      */
1291
1292     /* ... omitted code ...
1293      */
1294
1295     /* ... omitted code ...
1296      */
1297
1298     /* ... omitted code ...
1299      */
1300
1301     /* ... omitted code ...
1302      */
1303
1304     /* ... omitted code ...
1305      */
1306
1307     /* ... omitted code ...
1308      */
1309
1310     /* ... omitted code ...
1311      */
1312
1313     /* ... omitted code ...
1314      */
1315
1316     /* ... omitted code ...
1317      */
1318
1319     /* ... omitted code ...
1320      */
1321
1322     /* ... omitted code ...
1323      */
1324
1325     /* ... omitted code ...
1326      */
1327
1328     /* ... omitted code ...
1329      */
1330
1331     /* ... omitted code ...
1332      */
1333
1334     /* ... omitted code ...
1335      */
1336
1337     /* ... omitted code ...
1338      */
1339
1340     /* ... omitted code ...
1341      */
1342
1343     /* ... omitted code ...
1344      */
1345
1346     /* ... omitted code ...
1347      */
1348
1349     /* ... omitted code ...
1350      */
1351
1352     /* ... omitted code ...
1353      */
1354
1355     /* ... omitted code ...
1356      */
1357
1358     /* ... omitted code ...
1359      */
1360
1361     /* ... omitted code ...
1362      */
1363
1364     /* ... omitted code ...
1365      */
1366
1367     /* ... omitted code ...
1368      */
1369
1370     /* ... omitted code ...
1371      */
1372
1373     /* ... omitted code ...
1374      */
1375
1376     /* ... omitted code ...
1377      */
1378
1379     /* ... omitted code ...
1380      */
1381
1382     /* ... omitted code ...
1383      */
1384
1385     /* ... omitted code ...
1386      */
1387
1388     /* ... omitted code ...
1389      */
1390
1391     /* ... omitted code ...
1392      */
1393
1394     /* ... omitted code ...
1395      */
1396
1397     /* ... omitted code ...
1398      */
1399
1400     /* ... omitted code ...
1401      */
1402
1403     /* ... omitted code ...
1404      */
1405
1406     /* ... omitted code ...
1407      */
1408
1409     /* ... omitted code ...
1410      */
1411
1412     /* ... omitted code ...
1413      */
1414
1415     /* ... omitted code ...
1416      */
1417
1418     /* ... omitted code ...
1419      */
1420
1421     /* ... omitted code ...
1422      */
1423
1424     /* ... omitted code ...
1425      */
1426
1427     /* ... omitted code ...
1428      */
1429
1430     /* ... omitted code ...
1431      */
1432
1433     /* ... omitted code ...
1434      */
1435
1436     /* ... omitted code ...
1437      */
1438
1439     /* ... omitted code ...
1440      */
1441
1442     /* ... omitted code ...
1443      */
1444
1445     /* ... omitted code ...
1446      */
1447
1448     /* ... omitted code ...
1449      */
1450
1451     /* ... omitted code ...
1452      */
1453
1454     /* ... omitted code ...
1455      */
1456
1457     /* ... omitted code ...
1458      */
1459
1460     /* ... omitted code ...
1461      */
1462
1463     /* ... omitted code ...
1464      */
1465
1466     /* ... omitted code ...
1467      */
1468
1469     /* ... omitted code ...
1470      */
1471
1472     /* ... omitted code ...
1473      */
1474
1475     /* ... omitted code ...
1476      */
1477
1478     /* ... omitted code ...
1479      */
1480
1481     /* ... omitted code ...
1482      */
1483
1484     /* ... omitted code ...
1485      */
1486
1487     /* ... omitted code ...
1488      */
1489
1490     /* ... omitted code ...
1491      */
1492
1493     /* ... omitted code ...
1494      */
1495
1496     /* ... omitted code ...
1497      */
1498
1499     /* ... omitted code ...
1500      */
1501
1502     /* ... omitted code ...
1503      */
1504
1505     /* ... omitted code ...
1506      */
1507
1508     /* ... omitted code ...
1509      */
1510
1511     /* ... omitted code ...
1512      */
1513
1514     /* ... omitted code ...
1515      */
1516
1517     /* ... omitted code ...
1518      */
1519
1520     /* ... omitted code ...
1521      */
1522
1523     /* ... omitted code ...
1524      */
1525
1526     /* ... omitted code ...
1527      */
1528
1529     /* ... omitted code ...
1530      */
1531
1532     /* ... omitted code ...
1533      */
1534
1535     /* ... omitted code ...
1536      */
1537
1538     /* ... omitted code ...
1539      */
1540
1541     /* ... omitted code ...
1542      */
1543
1544     /* ... omitted code ...
1545      */
1546
1547     /* ... omitted code ...
1548      */
1549
1550     /* ... omitted code ...
1551      */
1552
1553     /* ... omitted code ...
1554      */
1555
1556     /* ... omitted code ...
1557      */
1558
1559     /* ... omitted code ...
1560      */
1561
1562     /* ... omitted code ...
1563      */
1564
1565     /* ... omitted code ...
1566      */
1567
1568     /* ... omitted code ...
1569      */
1570
1571     /* ... omitted code ...
1572      */
1573
1574     /* ... omitted code ...
1575      */
1576
1577     /* ... omitted code ...
1578      */
1579
1580     /* ... omitted code ...
1581      */
1582
1583     /* ... omitted code ...
1584      */
1585
1586     /* ... omitted code ...
1587      */
1588
1589     /* ... omitted code ...
1590      */
1591
1592     /* ... omitted code ...
1593      */
1594
1595     /* ... omitted code ...
1596      */
1597
1598     /* ... omitted code ...
1599      */
1600
1601     /* ... omitted code ...
1602      */
1603
1604     /* ... omitted code ...
1605      */
1606
1607     /* ... omitted code ...
1608      */
1609
1610     /* ... omitted code ...
1611      */
1612
1613     /* ... omitted code ...
1614      */
1615
1616     /* ... omitted code ...
1617      */
1618
1619     /* ... omitted code ...
1620      */
1621
1622     /* ... omitted code ...
1623      */
1624
1625     /* ... omitted code ...
1626      */
1627
1628     /* ... omitted code ...
1629      */
1630
1631     /* ... omitted code ...
1632      */
1633
1634     /* ... omitted code ...
1635      */
1636
1637     /* ... omitted code ...
1638      */
1639
1640     /* ... omitted code ...
1641      */
1642
1643     /* ... omitted code ...
1644      */
1645
1646     /* ... omitted code ...
1647      */
1648
1649     /* ... omitted code ...
1650      */
1651
1652     /* ... omitted code ...
1653      */
1654
1655     /* ... omitted code ...
1656      */
1657
1658     /* ... omitted code ...
1659      */
1660
1661     /* ... omitted code ...
1662      */
1663
1664     /* ... omitted code ...
1665      */
1666
1667     /* ... omitted code ...
1668      */
1669
1670     /* ... omitted code ...
1671      */
1672
1673     /* ... omitted code ...
1674      */
1675
1676     /* ... omitted code ...
1677      */
1678
1679     /* ... omitted code ...
1680      */
1681
1682     /* ... omitted code ...
1683      */
1684
1685     /* ... omitted code ...
1686      */
1687
1688     /* ... omitted code ...
1689      */
1690
1691     /* ... omitted code ...
1692      */
1693
1694     /* ... omitted code ...
1695      */
1696
1697     /* ... omitted code ...
1698      */
1699
1700     /* ... omitted code ...
1701      */
1702
1703     /* ... omitted code ...
1704      */
1705
1706     /* ... omitted code ...
1707      */
1708
1709     /* ... omitted code ...
1710      */
1711
1712     /* ... omitted code ...
1713      */
1714
1715     /* ... omitted code ...
1716      */
1717
1718     /* ... omitted code ...
1719      */
1720
1721     /* ... omitted code ...
1722      */
1723
1724     /* ... omitted code ...
1725      */
1726
1727     /* ... omitted code ...
1728      */
1729
1730     /* ... omitted code ...
1731      */
1732
1733     /* ... omitted code ...
1734      */
1735
1736     /* ... omitted code ...
1737      */
1738
1739     /* ... omitted code ...
1740      */
1741
1742     /* ... omitted code ...
1743      */
1744
1745     /* ... omitted code ...
1746      */
1747
1748     /* ... omitted code ...
1749      */
1750
1751     /* ... omitted code ...
1752      */
1753
1754     /* ... omitted code ...
1755      */
1756
1757     /* ... omitted code ...
1758      */
1759
1760     /* ... omitted code ...
1761      */
1762
1763     /* ... omitted code ...
1764      */
1765
1766     /* ... omitted code ...
1767      */
1768
1769     /* ... omitted code ...
1770      */
1771
1772     /* ... omitted code ...
1773      */
1774
1775     /* ... omitted code ...
1776      */
1777
1778     /* ... omitted code ...
1779      */
1780
1781     /* ... omitted code ...
1782      */
1783
1784     /* ... omitted code ...
1785      */
1786
1787     /* ... omitted code ...
1788      */
1789
1790     /* ... omitted code ...
1791      */
1792
1793     /* ... omitted code ...
1794      */
1795
1796     /* ... omitted code ...
1797      */
1798
1799     /* ... omitted code ...
1800      */
1801
1802     /* ... omitted code ...
1803      */
1804
1805     /* ... omitted code ...
1806      */
1807
1808     /* ... omitted code ...
1809      */
1810
1811     /* ... omitted code ...
1812      */
1813
1814     /* ... omitted code ...
1815      */
1816
1817     /* ... omitted code ...
1818      */
1819
1820     /* ... omitted code ...
1821      */
1822
1823     /* ... omitted code ...
1824      */
1825
1826     /* ... omitted code ...
1827      */
1828
1829     /* ... omitted code ...
1830      */
1831
1832     /* ... omitted code ...
1833      */
1834
1835     /* ... omitted code ...
1836      */
1837
1838     /* ... omitted code ...
1839      */
1840
1841     /* ... omitted code ...
1842      */
1843
1844     /* ... omitted code ...
1845      */
1846
1847     /* ... omitted code ...
1848      */
1849
1850     /* ... omitted code ...
1851      */
1852
1853     /* ... omitted code ...
1854      */
1855
1856     /* ... omitted code ...
1857      */
1858
1859     /* ... omitted code ...
1860      */
1861
1862     /* ... omitted code ...
1863      */
1864
1865     /* ... omitted code ...
1866      */
1867
1868     /* ... omitted code ...
1869      */
1870
1871     /* ... omitted code ...
1872      */
1873
1874     /* ... omitted code ...
1875      */
1876
1877     /* ... omitted code ...
1878      */
1879
1880     /* ... omitted code ...
1881      */
1882
1883     /* ... omitted code ...
1884      */
1885
1886     /* ... omitted code ...
1887      */
1888
1889     /* ... omitted code ...
1890      */
1891
1892     /* ... omitted code ...
1893      */
1894
1895     /* ... omitted code ...
1896      */
1897
1898     /* ... omitted code ...
1899      */
1900
1901     /* ... omitted code ...
1902      */
1903
1904     /* ... omitted code ...
1905      */
1906
1907     /* ... omitted code ...
1908      */
1909
1910     /* ... omitted code ...
1911      */
1912
1913     /* ... omitted code ...
1914      */
1915
1916     /* ... omitted code ...
1917      */
1918
1919     /* ... omitted code ...
1920      */
1921
1922     /* ... omitted code ...
1923      */
1924
1925     /* ... omitted code ...
1926      */
1927
1928
```

new/usr/src/cmd/fs.d/nfs/statd/sm_svc.c

1

```

540     struct rlimit rl;
541     int mode;
542     int sz;
543     int pipe_fd = -1;
544     int ret;
545     int commmaxrec = RPC_MAXDATASIZE;
546     struct netconfig *nconf;
547     NCONF_HANDLE *nc;

549     addrix = 0;
550     pathix = 0;

552     (void) gethostname(hostname, MAXHOSTNAMELEN);
553     if (init_hostname() < 0)
554         exit(1);

556     ret = nfs_smf_get_iprop("statd_port", &statd_port,
557                             DEFAULT_INSTANCE, SCF_TYPE_INTEGER, STATD);
558     if (ret != SA_OK) {
559         syslog(LOG_ERR, "Reading of statd_port from SMF "
560               "failed, using default value");
561     }

563     while ((c = getopt(argc, argv, "Dd:a:G:p:rU:")) != EOF)
564     while ((c = getopt(argc, argv, "Dd:a:G:p:rU:")) != EOF)
565         switch (c) {
566             case 'd':
567                 (void) sscanf(optarg, "%d", &debug);
568                 break;
569             case 'D':
570                 choice = 1;
571                 break;
572             case 'a':
573                 if (addrix < host_name_count) {
574                     if (strcmp(hostname, optarg) != 0) {
575                         sz = strlen(optarg);
576                         if (sz < MAXHOSTNAMELEN) {
577                             host_name[addrix] =
578                                 (char *)xmalloc(sz+1);
579                             if (host_name[addrix] !=
580                                 NULL) {
581                                 (void) sscanf(optarg, "%s",
582                               host_name[addrix]);
583                                 addrix++;
584                             }
585                             (void) fprintf(stderr,
586                               "statd: -a name of host is too long.\n");
587                         }
588                     } else
589                         (void) fprintf(stderr,
590                           "statd: -a exceeding maximum hostnames\n");
591                 }
592             break;
593             case 'U':
594                 (void) sscanf(optarg, "%d", &daemon_uid);
595                 break;
596             case 'G':
597                 (void) sscanf(optarg, "%d", &daemon_gid);
598                 break;
599             case 'p':
600                 if (strlen(optarg) < MAXPATHLEN) {
601                     /* If the path_name array has not yet      */
602                     /* been malloc'ed, do that. The array      */
603                     /* should be big enough to hold all of the */
604                     /* -p options we might have. An upper      */
605                     /* bound on the number of -p options is      */
606                     /* 1000. This is probably conservative       */
607                     /* given the size of the largest hostnames   */
608                     /* I've seen, but it's better to be safe than */
609                     /* sorry.                                     */
610                     sz = strlen(optarg);
611                     if (sz < MAXPATHLEN) {
612                         host_name[addrix] =
613                             (char *)xmalloc(sz+1);
614                         if (host_name[addrix] !=
615                             NULL) {
616                             (void) sscanf(optarg, "%s",
617                               host_name[addrix]);
618                             addrix++;
619                         }
620                     }
621                 }
622             break;
623         }
624     }
625 
```

new/usr/src/cmd/fs.d/nfs/statd/sm_svc.c

```

605     /* argc/2, because each -p option consumes */
606     /* two arguments. Here the upper bound */
607     /* is supposing that all the command line */
608     /* arguments are -p options, which would */
609     /* actually never be the case. */
610
611     if (path_name == NULL) {
612         size_t sz = (argc/2) * sizeof (char *);
613
614         path_name = (char **)malloc(sz);
615         if (path_name == NULL) {
616             (void) fprintf(stderr,
617                           "statd: malloc failed\n");
618             exit(1);
619         }
620         (void) memset(path_name, 0, sz);
621     }
622     path_name[pathix] = optarg;
623     pathix++;
624 } else {
625     (void) fprintf(stderr,
626                   "statd: -p pathname is too long.\n");
627 }
628 break;
629
630 case 'P':
631     (void) sscanf(optarg, "%d", &statd_port);
632     if (statd_port < 1 || statd_port > UINT16_MAX) {
633         (void) fprintf(stderr,
634                       "statd: -P port invalid.\n");
635         statd_port = 0;
636     }
637     break;
638 case 'r':
639     regfiles_only = 1;
640     break;
641 default:
642     (void) fprintf(stderr,
643                   "statd [-d level] [-D]\n");
644     return (1);
645 }
646
647 if (choice == 0) {
648     (void) strcpy(statd_home, home0);
649     (void) strcpy(CURRENT, current0);
650     (void) strcpy(BACKUP, backup0);
651     (void) strcpy(STATE, state0);
652 } else {
653     (void) strcpy(statd_home, home1);
654     (void) strcpy(CURRENT, current1);
655     (void) strcpy(BACKUP, backup1);
656     (void) strcpy(STATE, state1);
657 }
658
659 if (debug)
660     (void) printf("debug is on, create entry: %s, %s, %s\n",
661                   CURRENT, BACKUP, STATE);
662
663 if (getrlimit(RLIMIT_NOFILE, &rl))
664     (void) printf("statd: getrlimit failed. \n");
665
666 /* Set maxfdlimit current soft limit */
667 rl.rlim_cur = rl.rlim_max;
668 if (setrlimit(RLIMIT_NOFILE, &rl) != 0)
669     syslog(LOG_ERR, "statd: unable to set RLIMIT_NOFILE to %d\n",
670           rl.rlim_cur);
671
672 (void) enable_extended_FILE_stdio(-1, -1);

```

```

671     if (!debug) {
672         pipe_fd = daemonize_init();
673
674         openlog("statd", LOG_PID, LOG_DAEMON);
675     }
676
677     (void) _create_daemon_lock(STATD, daemon_uid, daemon_gid);
678
679     /* establish our lock on the lock file and write our pid to it.
680      * exit if some other process holds the lock, or if there's any
681      * error in writing/locking the file.
682      */
683     ppid = _enter_daemon_lock(STATD);
684     switch (ppid) {
685     case 0:
686         break;
687     case -1:
688         syslog(LOG_ERR, "error locking for %s: %s",
689               strerror(errno));
690         exit(2);
691     default:
692         /* daemon was already running */
693         exit(0);
694     }
695
696     mutex_init(&merges_lock, USYNC_THREAD, NULL);
697     cond_init(&merges_cond, USYNC_THREAD, NULL);
698     in_merges = B_TRUE;
699
700     /*
701      * Create thr_statd_merges() thread to populate the host_name list
702      * asynchronously.
703      */
704     if (thr_create(NULL, 0, (void * (*)(void *))thr_statd_merges, NULL,
705                   THR_DETACHED, NULL) != 0) {
706         syslog(LOG_ERR, "statd: unable to create thread for "
707               "'thr_statd_merges().'");
708         exit(1);
709     }
710
711     /*
712      * Set to automatic mode such that threads are automatically
713      * created
714      */
715     mode = RPC_SVC_MT_AUTO;
716     if (!rpc_control(RPC_SVC_MTMODE_SET, &mode)) {
717         syslog(LOG_ERR,
718               "statd:unable to set automatic MT mode.");
719         exit(1);
720     }
721
722     /*
723      * Set non-blocking mode and maximum record size for
724      * connection oriented RPC transports.
725      */
726     if (!rpc_control(RPC_SVC_CONNMAXREC_SET, &connmaxrec)) {
727         syslog(LOG_INFO, "unable to set maximum RPC record size");
728     }
729
730     /*
731      * Enumerate network transports and create service listeners
732      * as appropriate for each.
733      */
734     if ((nc = setnetconfig()) == NULL) {
735         syslog(LOG_ERR, "setnetconfig failed: %m");
736         return (-1);

```

```

639     if (!svc_create(sm_prog_1, SM_PROG, SM_VERS, "netpath")) {
640         syslog(LOG_ERR, "statd: unable to create (SM_PROG, SM_VERS) "
641               "for netpath.");
642         exit(1);
643     }
644     while ((nconf = getnetconfig(nc)) != NULL) {
645
646         /*
647          * Skip things like tpi_raw, invisible...
648          */
649         if ((nconf->nc_flag & NC_VISIBLE) == 0)
650             continue;
651         if (nconf->nc_semantics != NC_TPI_CLTS &&
652             nconf->nc_semantics != NC_TPI_COTS &&
653             nconf->nc_semantics != NC_TPI_COTS_ORD)
654             continue;
655
656         sm_svc_tp_create(nconf);
657         if (!svc_create(sm_prog_1, NSM_ADDR_PROGRAM, NSM_ADDR_V1, "netpath")) {
658             syslog(LOG_ERR, "statd: unable to create (NSM_ADDR_PROGRAM, "
659                   "NSM_ADDR_V1) for netpath.");
660         }
661         (void) endnetconfig(nc);
662
663         /*
664          * Make sure /var/statmon and any alternate (-p) statmon
665          * directories exist and are owned by daemon. Then change our uid
666          * to daemon. The uid change is to prevent attacks against local
667          * daemons that trust any call from a local root process.
668          */
669         set_statmon_owner();
670
671         /*
672          *
673          * statd now runs as a daemon rather than root and can not
674          * dump core under / because of the permission. It is
675          * important that current working directory of statd be
676          * changed to writable directory /var/statmon so that it
677          * can dump the core upon the receipt of the signal.
678          * One still need to set allow_setuid_core to non-zero in
679          * /etc/system to get the core dump.
680          */
681
682         if (chdir(statd_home) < 0) {
683             syslog(LOG_ERR, "can't chdir %s: %m", statd_home);
684             exit(1);
685         }
686
687         copy_client_names();
688
689         rwlock_init(&thr_rwlock, USYNC_THREAD, NULL);
690         mutex_init(&crash_lock, USYNC_THREAD, NULL);
691         mutex_init(&name_addrlock, USYNC_THREAD, NULL);
692         cond_init(&retrywait, USYNC_THREAD, NULL);
693         sm_inithash();
694         die = 0;
695
696         /*
697          * This variable is set to ensure that an sm_crash
698          * request will not be done at the same time
699          * when a statd_init is being done, since sm_crash
700          * can reset some variables that statd_init will be using.
701          */
702         in_crash = 1;
703         statd_init();
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2598
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2
```

```
797     /*  
798      * statd is up and running as far as we are concerned.  
799      */  
800     daemonize_fini(pipe_fd);  
  
802     if (debug)  
803         (void) printf("Starting svc_run\n");  
804     svc_run();  
805     syslog(LOG_ERR, "statd: svc_run returned\n");  
806     /* NOTREACHED */  
807     thr_exit((void *)1);  
808     return (0);  
  
810 }  
unchanged portion omitted
```

```
*****
6447 Mon Jul 17 16:15:17 2017
new/usr/src/cmd/fs.d/nfs/svc/server.xml
7577 mounted support to run on a fixed port
Portions contributed by: Paul Dagnelie <pcd@delphix.com>
Reviewed by: Evan Layton <evan.layton@nexenta.com>
Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>
*****
1 <?xml version="1.0"?>
2 <!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
3 <!--
5 CDDL HEADER START
7 The contents of this file are subject to the terms of the
8 Common Development and Distribution License (the "License").
9 You may not use this file except in compliance with the License.
11 You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
12 or http://www.opensolaris.org/os/licensing.
13 See the License for the specific language governing permissions
14 and limitations under the License.
16 When distributing Covered Code, include this CDDL HEADER in each
17 file and include the License file at usr/src/OPENSOLARIS.LICENSE.
18 If applicable, add the following below this CDDL HEADER, with the
19 fields enclosed by brackets "[]" replaced with your own identifying
20 information: Portions Copyright [yyyy] [name of copyright owner]
22 CDDL HEADER END
24 Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.
25 Copyright (c) 2012 by Delphix. All rights reserved.
26 Copyright 2014 Nexenta Systems, Inc. All rights reserved
27 Copyright 2016 Hans Rosenthal <rosenthal@grumpf.hope-2000.org>
28 Copyright (c) 2012, 2014 by Delphix. All rights reserved.
30 NOTE: This service manifest is not editable; its contents will
31 be overwritten by package or patch operations, including
32 operating system upgrade. Make customizations in a different
33 file.
35 Note: if this service is modified to consist of anything other
36 than a single instance named 'default', you must make changes to
37 $SRC/head/rpcsvc/daemon_utils.h and libnsl:open_daemon_lock().
38 -->
40 <service_bundle type='manifest' name='SUNWnfssr:nfs-server'>
42 <service
43   name='network/nfs/server'
44   type='service'
45   version='2'>
47   <dependency name='network'
48     grouping='require_any'
49     restart_on='error'
50     type='service'>
51     <service_fMRI value='svc:/milestone/network' />
52   </dependency>
54   <dependency name='nlockmgr'
55     grouping='require_all'
56     restart_on='error'
57     type='service'>
58     <service_fMRI value='svc:/network/nfs/nlockmgr' />
```

```
59   </dependency>
61   <dependency name='mapid'
62     grouping='optional_all'
63     restart_on='error'
64     type='service'>
65     <service_fMRI value='svc:/network/nfs/mapid' />
66   </dependency>
68   <dependency name='rpcbind'
69     grouping='require_all'
70     restart_on='restart'
71     type='service'>
72     <service_fMRI value='svc:/network/rpc/bind' />
73   </dependency>
75   <dependency name='keyserv'
76     grouping='optional_all'
77     restart_on='none'
78     type='service'>
79     <service_fMRI value='svc:/network/rpc/keyserv' />
80   </dependency>
82   <dependency name='gss'
83     grouping='optional_all'
84     restart_on='none'
85     type='service'>
86     <service_fMRI value='svc:/network/rpc/gss' />
87   </dependency>
89   <dependency name='share-group'
90     grouping='optional_all'
91     restart_on='none'
92     type='service'>
93     <service_fMRI value='svc:/network/shares/group' />
94   </dependency>
96   <dependency name='reparse'
97     grouping='optional_all'
98     restart_on='none'
99     type='service'>
100    <service_fMRI value='svc:/system/filesystem/reparse' />
101  </dependency>
103  <!-- Must have all local filesystems mounted before we share them -->
104  <dependency name='filesystem-local'
105    grouping='require_all'
106    restart_on='error'
107    type='service'>
108    <service_fMRI value='svc:/system/filesystem/local' />
109  </dependency>
111  <dependent
112    name='nfs-server_multi-user-server'
113    grouping='optional_all'
114    restart_on='none'>
115    <service_fMRI value='svc:/milestone/multi-user-server' />
116  </dependent>
118  <!--
119    The shareall done as part of the nfs-server method may take a
120    long time, as the contents of dfstab can be arbitrarily large.
121    Set the timeout appropriately for both 'start' and 'refresh'.
122  -->
123  <exec_method
124    type='method'
```

```

125      name='start'
126      exec='/lib/svc/method/nfs-server %m'
127      timeout_seconds='3600' />
128
129      <exec_method
130          type='method'
131          name='refresh'
132          exec='/lib/svc/method/nfs-server %m'
133          timeout_seconds='3600' />
134
135      <!--
136          The stop method runs unshareall as well as up to a 10 second
137          sleep to do graceful versus forceful shutdown of daemons. Set
138          the timeout appropriately.
139      -->
140      <exec_method
141          type='method'
142          name='stop'
143          exec='/lib/svc/method/nfs-server %m %{restarter/contract}'
144          timeout_seconds='3600' />
145
146      <property_group name='application' type='framework'>
147          <stability value='Evolving' />
148          <propval name='auto_enable' type='boolean' value='true' />
149      </property_group>
150
151      <property_group name='firewall_context' type='com.sun/fw_definition'>
152          <propval name='name' type='astring' value='nfsd' />
153          <propval name='ipf_method' type='astring'
154              value='/lib/svc/method/nfs-server ipfilter' />
155      </property_group>
156
157      <property_group name='firewall_config' type='com.sun/fw_configuration'>
158          <propval name='policy' type='astring' value='use_global' />
159          <propval name='block_policy' type='astring'
160              value='use_global' />
161          <propval name='apply_to' type='astring' value=''/>
162          <propval name='apply_to_6' type='astring' value=''/>
163          <propval name='exceptions' type='astring' value=''/>
164          <propval name='exceptions_6' type='astring' value=''/>
165          <propval name='target' type='astring' value=''/>
166          <propval name='target_6' type='astring' value=''/>
167          <propval name='value_authorization' type='astring'
168              value='solaris.smf.value.firewall.config' />
169      </property_group>
170      <instance name='default' enabled='false'>
171          <property_group name='nfs-props' type='com.oracle.nfs.props'>
172              <propval name='device' type='astring' value=''/>
173              <propval name='listen_backlog' type='integer' value='32' />
174              <propval name='max_connections' type='integer' value=' -1' />
175              <propval name='protocol' type='astring' value='ALL' />
176              <propval name='server_delegation' type='astring' value='on' />
177              <propval name='server_versmax' type='integer' value='4' />
178              <propval name='server_versmin' type='integer' value='2' />
179              <propval name='servers' type='integer' value='1024' />
180              <propval name='mountd_listen_backlog' type='integer' value='64' />
181              <propval name='mountd_max_threads' type='integer' value='16' />
182              <propval name='mountd_port' type='integer' value='0' />
183          </property_group>
184      </instance>
185
186      <stability value='Stable' />
187
188      <template>
189          <common_name>
190              <loctext xml:lang='C'>
```

```

191          NFS server
192      </loctext>
193      </common_name>
194      <documentation>
195          <manpage title='nfsd' section='1M'
196              manpath='/usr/share/man' />
197      </documentation>
198      </template>
199  </service>
200 </service_bundle>
```

```
new/usr/src/cmd/fs.d/nfs/svc/status.xml
```

```
*****
3247 Mon Jul 17 16:15:17 2017
new/usr/src/cmd/fs.d/nfs/svc/status.xml
7569 statd support to run on a fixed port
Portions contributed by: Paul Dagnelie <pcd@delphix.com>
Reviewed by: Evan Layton <evan.layton@nexenta.com>
Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>
*****
1 <?xml version="1.0"?>
2 <!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
3 <!--
4 Copyright 2009 Sun Microsystems, Inc. All rights reserved.
5 Use is subject to license terms.
7 CDDL HEADER START
9 The contents of this file are subject to the terms of the
10 Common Development and Distribution License (the "License").
11 You may not use this file except in compliance with the License.
13 You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
14 or http://www.opensolaris.org/os/licensing.
15 See the License for the specific language governing permissions
16 and limitations under the License.
18 When distributing Covered Code, include this CDDL HEADER in each
19 file and include the License file at usr/src/OPENSOLARIS.LICENSE.
20 If applicable, add the following below this CDDL HEADER, with the
21 fields enclosed by brackets "[]" replaced with your own identifying
22 information: Portions Copyright [yyyy] [name of copyright owner]
24 CDDL HEADER END
26 NOTE: This service manifest is not editable; its contents will
27 be overwritten by package or patch operations, including
28 operating system upgrade. Make customizations in a different
29 file.
31 Note: if this service is modified to consist of anything other
32 than a single instance named 'default', you must make changes to
33 $SRC/head/rpcsvc/daemon_utils.h and libnsl:open_daemon_lock().
34 -->
36 <!-- Copyright (c) 2016 by Delphix. All rights reserved. -->
38 <service_bundle type='manifest' name='SUNWnfscr:nfs-status'>
40 <service
41   name='network/nfs/status'
42   type='service'
43   version='1'>
43   <create_default_instance enabled='false' />
45     <single_instance />
45   <dependency name='network'
46     grouping='require_any'
47     restart_on='error'
48     type='service'>
49     <service_fmri value='svc:/milestone/network' />
50   </dependency>
52   <dependency name='rpcbind'
53     grouping='require_all'
54     restart_on='restart'>
```

```
1
```

```
new/usr/src/cmd/fs.d/nfs/svc/status.xml
55       type='service'>
56         <service_fmri value='svc:/network/rpc/bind' />
57       </dependency>
59       <dependency name='filesystem-local'
60         grouping='require_all'
61         restart_on='error'
62         type='service'>
63         <service_fmri value='svc:/system/filesystem/local' />
64       </dependency>
66       <exec_method
67         type='method'
68         name='start'
69         exec='/usr/lib/nfs/statd'
70         timeout_seconds='60' />
72       <exec_method
73         type='method'
74         name='stop'
75         exec=':kill'
76         timeout_seconds='60' />
78       <property_group name='firewall_context' type='com.sun,fw_definition'>
79         <propval name='isrpc' type='boolean' value='true' />
80         <propval name='name' type='astring' value='status' />
81         <propval name='ipf_method' type='astring'
82           value='/lib/svc/method/nfs-server ipfilter' />
83       </property_group>
85       <property_group name='application' type='framework'>
86         <stability value='Evolving' />
87         <propval name='auto_enable' type='boolean' value='true' />
88       </property_group>
90       <instance name='default' enabled='false'>
91         <property_group name='nfs-props' type='com.oracle.nfs,props'>
92           <propval name='statd_port' type='integer' value='0' />
93         </property_group>
94       </instance>
96         <stability value='Stable' />
98       <template>
99         <common_name>
100           <loctext xml:lang='C'>
101             NFS status monitor
102           </loctext>
103         </common_name>
104         <documentation>
105           <manpage title='statd' section='1M'
106             manpath='/usr/share/man' />
107         </documentation>
108       </template>
109     </service>
111   </service_bundle>
```

```
2
```

```
*****
1269 Mon Jul 17 16:15:17 2017
new/usr/src/cmd/fs.d/nfs/tests/Makefile
8330 Add svc_tp_create_addr to libnsl
Reviewed by: Paul Dagnelie <pcd@delphix.com>
Reviewed by: Evan Layton <evan.layton@nexenta.com>
Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>
*****
```

```
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, Version 1.0 only
6 # (the "License"). You may not use this file except in compliance
7 # with the License.
8 #
9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 # or http://www.opensolaris.org/os/licensing.
11 # See the License for the specific language governing permissions
12 # and limitations under the License.
13 #
14 # When distributing Covered Code, include this CDDL HEADER in each
15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 # If applicable, add the following below this CDDL HEADER, with the
17 # fields enclosed by brackets "[]" replaced with your own identifying
18 # information: Portions Copyright [yyyy] [name of copyright owner]
19 #
20 # CDDL HEADER END
21 #
22 #
23 # Copyright 2004 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # cmd/fs.d/nfs/tests/Makefile

28 FSTYPE= nfs
29 LIBPROG= test_svc_tp_create

31 include ../../Makefile.fstype

33 OBJS= $(LIBPROG).o
34 SRCS= $(LIBPROG).c

36 CFLAGS += $(CCVERBOSE)

38 LDLIBS += -lnsl -lsocket

40 # message catalog
41 catalog:

43 $(LIBPROG): $(OBJS)
44         $(LINK.c) -o $@ $(OBJS) $(LDLIBS)
45         $(POST_PROCESS)

47 lint: lint_SRCS

49 clean:
50     $(RM) $(LIBPROG).o
```

```
new/usr/src/cmd/fs.d/nfs/tests/test_svc_tp_create.c
*****
7518 Mon Jul 17 16:15:17 2017
new/usr/src/cmd/fs.d/nfs/tests/test_svc_tp_create.c
8330 Add svc_tp_create_addr to libnsl
Reviewed by: Paul Dagnelie <pcd@delphix.com>
Reviewed by: Evan Layton <evan.layton@nexenta.com>
Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>
*****
```

```
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22 * Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
23 * Copyright (c) 2012, 2016 by Delphix. All rights reserved.
24 * Copyright 2017 Nexenta Systems, Inc. All rights reserved.
25 */
26 /*
27 *      Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989 AT&T      */
28 /*      All Rights Reserved      */
29 /*
30 */
31 /*
32 * Portions of this source code were derived from Berkeley 4.3 BSD
33 * under license from the Regents of the University of California.
34 */
35 #include <stdio.h>
36 #include <stdlib.h>
37 #include <ctype.h>
38 #include <sys/types.h>
39 #include <string.h>
40 #include <syslog.h>
41 #include <sys/param.h>
42 #include <rpc/rpc.h>
43 #include <sys/stat.h>
44 #include <netconfig.h>
45 #include <netdir.h>
46 #include <sys/file.h>
47 #include <sys/time.h>
48 #include <sys/errno.h>
49 #include <rpcsvc/mount.h>
50
51 #include <signal.h>
52 #include <locale.h>
53 #include <unistd.h>
54 #include <errno.h>
55 #include <sys/socket.h>
56 #include <netinet/in.h>
```

1

new/usr/src/cmd/fs.d/nfs/tests/test_svc_tp_create.c

```
59 #include <arpa/inet.h>
60 #include <netdb.h>
62 #include <thread.h>
63 #include <assert.h>
65 #include <limits.h>
67 #define TESTPROG 987654
69 uint32_t test_vers_max = 2;
70 uint32_t test_vers_min = 1;
72 int debug;
73 int verbose;
74 int testd_port;
76 static void mysvc(struct svc_req *, SVCXPRT *);
77 static void bind2(void);
79 /*
80 * This function is called for each configured network type to
81 * bind and register our RPC service programs.
82 */
83 * On TCP or UDP, we want to bind TESTPROG on a specific port
84 * (when testd_port is specified) in which case we'll use the
85 * variant of svc_tp_create() that lets us pass a bind address.
86 */
87 static void
88 test_svc_tp_create(struct netconfig *nconf)
89 {
90     char port_str[8];
91     struct nd_hostserv hs;
92     struct nd_addrlist *al = NULL;
93     SVCXPRT *xprt = NULL;
94     rpcvers_t vers;
96     vers = test_vers_max;
98     /*
99      * If testd_port is set and this is an inet transport,
100     * bind this service on the specified port.
101     */
102     if (testd_port != 0 &&
103         (strcmp(nconf->nc_protofml, NC_INET) == 0 ||
104          strcmp(nconf->nc_protofml, NC_INET6) == 0)) {
105         int err;
106
107         sprintf(port_str, sizeof (port_str), "%u",
108                 (unsigned short)testd_port);
109
110         hs.h_host = HOST_SELF_BIND;
111         hs.h_serv = port_str;
112         err = netdir_getbyname((struct netconfig *)nconf, &hs, &al);
113         if (err == 0 && al != NULL) {
114             xprt = svc_tp_create_addr(mysvc, TESTPROG, vers,
115                                       nconf, al->n_addrs);
116             netdir_free(al, ND_ADDRLIST);
117         }
118         if (xprt == NULL) {
119             printf("testd: unable to create "
120                   "(TESTD,%d) on transport %s (port %d)\n",
121                   vers, nconf->nc_netid, testd_port);
122         }
123     /* fall-back to default bind */
124 }
```

2

```

125     if (xprt == NULL) {
126         /*
127          * Had testd_port=0, or non-inet transport,
128          * or the bind to a specific port failed.
129          */
130         /*
131         * Do a default bind.
132         */
133         xprt = svc_tp_create(mysvc, TESTPROG, vers, nconf);
134     }
135     if (xprt == NULL) {
136         printf("testd: unable to create "
137             "(TESTD,%d) on transport %s\n",
138             vers, nconf->nc_netid);
139         return;
140     }
141     /*
142      * Register additional versions on this transport.
143      */
144     while (--vers >= test_vers_min) {
145         if (!svc_reg(xprt, TESTPROG, vers, mysvc, nconf)) {
146             printf("testd: "
147                 "failed to register vers %d on %s\n",
148                 vers, nconf->nc_netid);
149     }
150 }

152 static void
153 test_svc_unreg(void)
154 {
155     rpcvers_t vers;
156
157     for (vers = test_vers_min; vers <= test_vers_max; vers++)
158         svc_unreg(TESTPROG, vers);
159 }

161 int
162 main(int argc, char *argv[])
163 {
164     int c;
165     bool_t exclbind = TRUE;
166     int tmp;
167     struct netconfig *nconf;
168     NCONF_HANDLE *nc;

169     while ((c = getopt(argc, argv, "dvp:")) != EOF) {
170         switch (c) {
171         case 'd':
172             debug++;
173             break;
174         case 'v':
175             verbose++;
176             break;
177         case 'p':
178             (void) sscanf(optarg, "%d", &tmp);
179             if (tmp < 1 || tmp > UINT16_MAX) {
180                 (void) fprintf(stderr,
181                     "testd: -P port invalid.\n");
182                 return (1);
183             }
184             testd_port = tmp;
185             break;
186         default:
187             fprintf(stderr, "usage: testd [-v] [-r]\n");
188             exit(1);
189     }

```

```

191     }
192     (void) setlocale(LC_ALL, "");
193
194 #if !defined(TEXT_DOMAIN)
195 #define TEXT_DOMAIN "SYS_TEST"
196 #endif
197 #endif
198     (void) textdomain(TEXT_DOMAIN);

200     /*
201      * Prevent our non-priv udp and tcp ports bound w/wildcard addr
202      * from being hijacked by a bind to a more specific addr.
203      */
204     if (!rpc_control(__RPC_SVC_EXCLBIND_SET, &exclbind)) {
205         fprintf(stderr, "warning: unable to set udp/tcp EXCLBIND\n");
206     }

207     if (testd_port < 0 || testd_port > UINT16_MAX) {
208         fprintf(stderr, "unable to use specified port\n");
209         exit(1);
210     }

211     /*
212      * Make sure to unregister any previous versions in case the
213      * user is reconfiguring the server in interesting ways.
214      */
215     test_svc_unreg();

216     /*
217      * Enumerate network transports and create service listeners
218      * as appropriate for each.
219      */
220     if ((nc = setnetconfig()) == NULL) {
221         perror("setnetconfig failed");
222         return (-1);
223     }
224     while ((nconf = getnetconfig(nc)) != NULL) {
225         /*
226          * Skip things like tpi_raw, invisible...
227          */
228         if ((nconf->nc_flag & NC_VISIBLE) == 0)
229             continue;
230         if (nconf->nc_semantics != NC_TPI_CLTS &&
231             nconf->nc_semantics != NC_TPI_COTS &&
232             nconf->nc_semantics != NC_TPI_COTS_ORD)
233             continue;
234
235         test_svc_tp_create(nconf);
236     }
237     (void) endnetconfig(nc);

238     /*
239      * XXX: Normally would call svc_run() here, but
240      * we just want to check our IP bindings.
241      */
242     if (testd_port != 0)
243         bind2();
244
245     if (debug) {
246         char sysbuf[100];
247
248         snprintf(sysbuf, sizeof (sysbuf),
249             "rpcinfo -p |grep %u", TESTPROG);
250         printf("x %s\n", sysbuf);
251         fflush(stdout);
252         system(sysbuf);
253     }

```

```

258     if (testd_port) {
259         snprintf(sysbuf, sizeof (sysbuf),
260                 "netstat -a -f inet -P udp |grep %u", testd_port);
261         printf("x %s\n", sysbuf);
262         fflush(stdout);
263         system(sysbuf);
264
265         snprintf(sysbuf, sizeof (sysbuf),
266                 "netstat -a -f inet -P tcp |grep %u", testd_port);
267         printf("x %s\n", sysbuf);
268         fflush(stdout);
269         system(sysbuf);
270     }
271 }
273 /* cleanup */
274 test_svc_unreg();
276 printf("%s complete\n", argv[0]);
277 return (0);
278 }

280 /*
281 * Server procedure switch routine
282 */
283 static void
284 mysvc(struct svc_req *rq, SVCXPRT *xprt)
285 {
286     switch (rq->rq_proc) {
287     case NULLPROC:
288         errno = 0;
289         (void)svc_sendreply(xprt, xdr_void, (char *)0);
290         return;
291
292     default:
293         svcerr_noproc(xprt);
294         return;
295     }
296 }
299 struct sockaddr_in addr;

301 /*
302 * The actual test: Try doing a 2nd bind with a specific IP.
303 * The exclusive wildcard bind should prevent this.
304 */
305 static void
306 bind2(void)
307 {
308     int ret;
309     int sock;

311     addr.sin_family = AF_INET;
312     addr.sin_port = htons(testd_port);
313     addr.sin_addr.s_addr = htonl(INADDR_LOOPBACK);

315     sock = socket(AF_INET, SOCK_STREAM, 0);
316     if (sock == -1) {
317         fprintf(stderr, "bind2 socket fail %s\n",
318                 strerror(errno));
319         exit(1);
320     }
322     ret = bind(sock, (struct sockaddr *)&addr, sizeof (addr));

```

```

323     if (ret == -1) {
324         fprintf(stderr, "bind2 bind fail %s (expected) PASS\n", strerror(errno));
325         close(sock);
326         return;
327     }
329     printf("Oh no, bind2 worked! test FAILED\n");
330     close(sock);
331     return;
332 }

```

```
new/usr/src/lib/libnsl/common/mapfile-vers
```

```
1
```

```
*****
```

```
14382 Mon Jul 17 16:15:18 2017
```

```
new/usr/src/lib/libnsl/common/mapfile-vers
```

```
8330 Add svc_tp_create_addr to libnsl
```

```
Reviewed by: Paul Dagnelie <pcd@delphix.com>
```

```
Reviewed by: Evan Layton <evan.layton@nexenta.com>
```

```
Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>
```

```
*****
```

```
1 #
```

```
2 # CDDL HEADER START
```

```
3 #
```

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

```
7 #
```

```
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE  
9 # or http://www.opensolaris.org/os/licensing.
```

```
10 # See the License for the specific language governing permissions  
11 # and limitations under the License.
```

```
12 #
```

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

```
18 #
```

```
19 # CDDL HEADER END
```

```
20 #
```

```
21 #
```

```
22 # Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
```

```
23 # Copyright 2017 Nexenta Systems, Inc. All rights reserved.
```

```
23 # Copyright 2014 Nexenta Systems, Inc. All rights reserved.
```

```
24 #
```

```
26 #
```

```
27 # MAPFILE HEADER START
```

```
28 #
```

```
29 # WARNING: STOP NOW. DO NOT MODIFY THIS FILE.
```

```
30 # Object versioning must comply with the rules detailed in
```

```
31 #
```

```
32 #     usr/src/lib/README.mapfiles
```

```
33 #
```

```
34 # You should not be making modifications here until you've read the most current  
35 # copy of that file. If you need help, contact a gatekeeper for guidance.
```

```
36 #
```

```
37 # MAPFILE HEADER END
```

```
38 #
```

```
40 $mapfile_version 2
```

```
42 SYMBOL_VERSION ILLUMOS_0.1 {      # Illumos additions
```

```
43     global:
```

```
44         svc_tp_create_addr;
```

```
45 } SUNW_1.10;
```

```
47 SYMBOL_VERSION SUNW_1.10 {      # SunOS 5.11 (Solaris 11)
```

```
48     global:
```

```
49         SUNW_1.10;
```

```
50 } SUNW_1.9.1;
```

```
unchanged portion omitted
```

new/usr/src/lib/libnsl/rpc/svc_generic.c

17805 Mon Jul 17 16:15:18 2017

new/usr/src/lib/libnsl/rpc/svc_generic.c

8330 Add svc_tp_create_addr to libnsl

Reviewed by: Paul Dagnelie <pcd@delphix.com>

Reviewed by: Evan Layton <evan.layton@nexenta.com>

Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>

```
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 2016 Nexenta Systems, Inc. All rights reserved.
24 * Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
25 * Copyright 2014 Nexenta Systems, Inc. All rights reserved.
26 */
27 /* Copyright (c) 1988 AT&T */
28 /* All Rights Reserved */

30 /*
31 * svc_generic.c, Server side for RPC.
32 */
33 */

35 #include "mt.h"
36 #include <stdlib.h>
37 #include <sys/socket.h>
38 #include <netinet/in.h>
39 #include <netinet/tcp.h>
40 #include <netinet/udp.h>
41 #include <inttypes.h>
42 #include "rpc_mt.h"
43 #include <stdio.h>
44 #include <rpc/rpc.h>
45 #include <sys/types.h>
46 #include <errno.h>
47 #include <syslog.h>
48 #include <rpc/nettype.h>
49 #include <malloc.h>
50 #include <string.h>
51 #include <stropts.h>
52 #include <tsol/label.h>
53 #include <nfs/nfs.h>
54 #include <nfs/nfs_acl.h>
55 #include <rpccsvc/mount.h>
56 #include <rpccsvc/nsm_addr.h>
57 #include <rpccsvc/rquot.h>
```

1

new/usr/src/lib/libnsl/rpc/svc_generic.c

```
58 #include <rpccsvc/sm_inter.h>
59 #include <rpccsvc/nlm_prot.h>
61 extern int __svc_vc_setflag(SVCXPRT *, int);
63 extern SVCXPRT *svc_dg_create_private(int, uint_t, uint_t);
64 extern SVCXPRT *svc_vc_create_private(int, uint_t, uint_t);
65 extern SVCXPRT *svc_fd_create_private(int, uint_t, uint_t);
67 extern bool_t __svc_add_to_xlist(SVCXPRT_LIST **, SVCXPRT *, mutex_t *);
68 extern void __svc_free_xlist(SVCXPRT_LIST **, mutex_t *);
70 extern bool_t __rpc_try_doors(const char *, bool_t *);
72 /*
73 * The highest level interface for server creation.
74 * It tries for all the nettokens in that particular class of token
75 * and returns the number of handles it can create and/or find.
76 *
77 * It creates a link list of all the handles it could create.
78 * If svc_create() is called multiple times, it uses the handle
79 * created earlier instead of creating a new handle every time.
80 */

82 /* VARIABLES PROTECTED BY xprtlist_lock: xprtlist */
84 SVCXPRT_LIST *_svc_xprtlist = NULL;
85 extern mutex_t xprtlist_lock;

87 static SVCXPRT * svc_tli_create_common(int, const struct netconfig *,
88 const struct t_bind *, uint_t, uint_t, boolean_t);
90 static SVCXPRT *svc_tp_create_bind(void (*dispatch)(),
91 const rpcprog_t, const rpcvers_t,
92 const struct netconfig *, const struct t_bind *);

94 boolean_t
95 is_multilevel(rpcprog_t progrum)
96 {
97     /* This is a list of identified multilevel service provider */
98     if ((progrum == MOUNTPROG) || (progrum == NFS_PROGRAM) ||
99         (progrum == NFS_ACL_PROGRAM) || (progrum == NLM_PROG) ||
100        (progrum == NSM_ADDR_PROGRAM) || (progrum == RQUOTAPROG) ||
101        (progrum == SM_PROG))
102         return (B_TRUE);
104     return (B_FALSE);
105 }
unchanged_portion_omitted

180 /*
181 * The high level interface to svc_tli_create().
182 * It tries to create a server for "nconf" and registers the service
183 * with the rpcbind.
184 * with the rpcbind. It calls svc_tli_create();
185 */
186 SVCXPRT *
187 svc_tp_create(void (*dispatch)(), const rpcprog_t progrum,
188 const rpcvers_t versnum, const struct netconfig *nconf)
189 {
190     return (svc_tp_create_bind(dispatch, progrum, versnum, nconf, NULL));
192 /*
193 * svc_tp_create_addr()
194 * Variant of svc_tp_create() that allows specifying just the
```

2

```

195 * the binding address, for convenience.
196 */
197 SVCXPRT *
198 svc_tp_create_addr(void (*dispatch)(), const rpcprog_t progrum,
199     const rpcvers_t versnum, const struct netconfig *nconf,
200     const struct netbuf *addr)
201 {
202     struct t_bind bind;
203     struct t_bind *bindp = NULL;
204
205     if (addr != NULL) {
206
207         bind.addr = *addr;
208         if (!rpc_control(_RPC_SVC_LISTENBKLOG_GET, &bind.qlen)) {
209             syslog(LOG_ERR,
210                 "svc_tp_create: can't get listen backlog");
211             return (NULL);
212         }
213         bindp = &bind;
214     }
215
216     /*
217      * When bindp == NULL, this is the same as svc_tp_create().
218      */
219     return (svc_tp_create_bind(dispatch, progrum, versnum,
220         nconf, bindp));
221 }
222
223 static SVCXPRT *
224 svc_tp_create_bind(void (*dispatch)(), const rpcprog_t progrum,
225     const rpcvers_t versnum, const struct netconfig *nconf,
226     const struct t_bind *bindaddr)
227 {
228     SVCXPRT *xprt;
229     boolean_t anon_mlp = B_FALSE;
230
231     if (nconf == NULL) {
232         (void) syslog(LOG_ERR, "svc_tp_create: invalid netconfig "
233             "structure for prog %d vers %d", progrum, versnum);
234         return (NULL);
235     }
236
237     /* Some programs need to allocate MLP for multilevel services */
238     if (is_system_labeled() && is_multilevel(progrum))
239         anon_mlp = B_TRUE;
240     xprt = svc_tli_create_common(RPC_ANYFD, nconf, bindaddr, 0, 0,
241         anon_mlp);
242     xprt = svc_tli_create_common(RPC_ANYFD, nconf, NULL, 0, 0, anon_mlp);
243     if (xprt == NULL)
244         return (NULL);
245
246     (void) rpcb_unset(progrum, versnum, (struct netconfig *)nconf);
247     if (svc_reg(xprt, progrum, versnum, dispatch, nconf) == FALSE) {
248         (void) syslog(LOG_ERR,
249             "svc_tp_create: Could not register prog %d vers %d on %s",
250             progrum, versnum, nconf->nc_netid);
251         SVC_DESTROY(xprt);
252         return (NULL);
253     }
254     return (xprt);
255 }


---


unchanged_portion_omitted
263 /*
264 * If fd is RPC_ANYFD, then it opens a fd for the given transport
265 * provider (nconf cannot be NULL then). If the t_state is T_UNBND and

```

```

266     * bindaddr is NON-NULL, it performs a t_bind using the bindaddr. For
267     * NULL bindaddr and Connection oriented transports, the value of qlen
268     * is set arbitrarily.
269
270     * If sendsz or recvsz are zero, their default values are chosen.
271     */
272 SVCXPRT *
273 svc_tli_create_common(const int ofd, const struct netconfig *nconf,
274     const struct t_bind *bindaddr, const uint_t sendsz,
275     const uint_t recvsz, boolean_t mlp_flag)
276 {
277     SVCXPRT *xprt = NULL;           /* service handle */
278     struct t_info tinfo;           /* transport info */
279     struct t_bind *tres = NULL;    /* bind info */
280     bool_t madefd = FALSE;        /* whether fd opened here */
281     int state;                   /* state of the transport provider */
282     int fd = ofd;
283
284     if (fd == RPC_ANYFD) {
285         if (nconf == NULL) {
286             (void) syslog(LOG_ERR,
287                 "svc_tli_create: invalid netconfig");
288             return (NULL);
289         }
290         fd = t_open(nconf->nc_device, O_RDWR, &tinfo);
291         if (fd == -1) {
292             char errorstr[100];
293
294             __tli_sys_strerror(errorstr, sizeof (errorstr),
295                 t_errno, errno);
296             (void) syslog(LOG_ERR, "svc_tli_create: could not open "
297                 "connection for %s: %s", nconf->nc_netid, errorstr);
298             return (NULL);
299         }
300         madefd = TRUE;
301         state = T_UNBND;
302     } else {
303         /*
304          * It is an open descriptor. Sync it & get the transport info.
305          */
306         if ((state = t_sync(fd)) == -1) {
307             char errorstr[100];
308
309             __tli_sys_strerror(errorstr, sizeof (errorstr),
310                 t_errno, errno);
311             (void) syslog(LOG_ERR,
312                 "svc_tli_create: could not do t_sync: %s",
313                 errorstr);
314             return (NULL);
315         }
316         if (t_getinfo(fd, &tinfo) == -1) {
317             char errorstr[100];
318
319             __tli_sys_strerror(errorstr, sizeof (errorstr),
320                 t_errno, errno);
321             (void) syslog(LOG_ERR, "svc_tli_create: could not get "
322                 "transport information: %s", errorstr);
323             return (NULL);
324         }
325         /*
326          * Enable options of returning the ip's for udp */
327         if (nconf) {
328             int ret = 0;
329             if (strcmp(nconf->nc_netid, "udp6") == 0) {
330                 ret = __rpc_tli_set_options(fd, IPPROTO_IPV6,
331                     IPV6_RECVPKTINFO, 1);
332                 if (ret < 0) {

```

new/usr/src/lib/libnsl/rpc/svc_generic.c

5

```

334                                     char errorstr[100];
335                                     __tli_sys_strerror(errorstr,
336                                         sizeof (errorstr), t_errno, errno);
337                                     (void) syslog(LOG_ERR,
338                                         "svc_tli_create: "
339                                         "IPV6_RECVPKTINFO(1): %s",
340                                         errorstr);
341                                     return (NULL);
342 } else if (strcmp(nconf->nc_netid, "udp") == 0) {
343     ret = __rpc_tli_set_options(fd, IPPROTO_IP,
344                                IP_RECVDSTADDR, 1);
345     if (ret < 0) {
346         char errorstr[100];
347
348         __tli_sys_strerror(errorstr,
349             sizeof (errorstr), t_errno, errno);
350         (void) syslog(LOG_ERR,
351             "svc_tli_create: "
352             "IP_RECVDSTADDR(1): %s", errorstr);
353
354     }
355 }
356 }
357 */
358 * If the fd is unbound, try to bind it.
359 * In any case, try to get its bound info in tres
360 */
361 /* LINTED pointer alignment */
362 tres = (struct t_bind *)t_alloc(fd, T_BIND, T_ADDR);
363 if (tres == NULL) {
364     (void) syslog(LOG_ERR, "svc_tli_create: No memory!");
365     goto freedata;
366 }
367
368 switch (state) {
369     bool_t tcp, exclbind;
370     case T_UNBND:
371         /* If this is a labeled system, then ask for an MLP */
372         if (is_system_labeled() &&
373             (strcmp(nconf->nc_protomly, NC_INET) == 0 ||
374              strcmp(nconf->nc_protomly, NC_INET6) == 0)) {
375             (void) __rpc_tli_set_options(fd, SOL_SOCKET,
376                                         SO_RECVUCRED, 1);
377             if (mlp_flag)
378                 (void) __rpc_tli_set_options(fd, SOL_SOCKET,
379                                         SO_ANON_MLP, 1);
380         }
381
382         if (bindaddr) {
383             /*
384             * Services that specify a bind address typically
385             * use a fixed service (IP port) so we need to set
386             * SO_REUSEADDR to prevent bind errors on restart.
387             * SO_EXCLBIND has the following properties
388             * - an fd bound to port P via IPv4 will prevent an IPv6
389             *   bind to port P (and vice versa)
390             * - an fd bound to a wildcard IP address for port P will
391             *   prevent a more specific IP address bind to port P
392             * (see {tcp,udp}.c for details)
393             *
394             * We use the latter property to prevent hijacking of RPC
395             * services that reside at non-privileged ports.
396         }
397     }
398 }
```

new/usr/src/lib/libnsl/rpc/svc_generic.c

```

/*
if (bindaddr->addr.len != 0)
    (void) __rpc_tli_set_options(fd, SOL_SOCKET,
        SO_REUSEADDR, 1);
tcp = nconf ? (strcmp(nconf->nc_proto, NC_TCP) == 0) : 0;
if (nconf &&
    (tcp || (strcmp(nconf->nc_proto, NC_UDP) == 0)) &&
    rpc_control(__RPC_SVC_EXCLBIND_GET, &exclbind)) {
    if (exclbind) {
        if (!__rpc_tli_set_options(fd, SOL_SOCKET,
            SO_EXCLBIND, 1) < 0) {
            syslog(LOG_ERR,
                "svc_tli_create: can't set EXCLBIND [netid='%s']",
                nconf->nc_netid);
            goto freedata;
        }
    }
}
if (bindaddr) {
    if (t_bind(fd, (struct t_bind *)bindaddr, tres) == -1) {
        __tli_sys_strerror(errorstr, sizeof (errorstr),
            t_errno, errno);
        (void) syslog(LOG_ERR,
            "svc_tli_create: could not bind: %s",
            errorstr);
        goto freedata;
    }
    /*
     * Should compare the addresses only if addr.len
     * was non-zero
     */
    if (bindaddr->addr.len &&
        (memcmp(bindaddr->addr.buf, tres->addr.buf,
            (int)tres->addr.len) != 0)) {
        (void) syslog(LOG_ERR, "svc_tli_create: could "
            "not bind to requested address: address "
            "mismatch");
        goto freedata;
    }
} else {
    if (rpc_control(__RPC_SVC_LSTNBKLOG_GET, &tres->qlen)
        == FALSE) {
        syslog(LOG_ERR,
            "svc_tli_create: can't get listen backlog");
        goto freedata;
    }
    tres->addr.len = 0;
    if (t_bind(fd, tres, tres) == -1) {
        char errorstr[100];
        __tli_sys_strerror(errorstr, sizeof (errorstr),
            t_errno, errno);
        (void) syslog(LOG_ERR,
            "svc_tli_create: could not bind: %s",
            errorstr);
        goto freedata;
    }
}

/*
 * If requested, set SO_EXCLBIND on each binding.
 *
 * SO_EXCLBIND has the following properties
 *   - an fd bound to port P via IPv4 will prevent an IPv6

```

```

439         * bind to port P (and vice versa)
440         * - an fd bound to a wildcard IP address for port P will
441         * prevent a more specific IP address bind to port P
442         * (see {tcp,udp}.c for details)
443         *
444         * We use the latter property to prevent hijacking of RPC
445         * services that reside at non-privileged ports.
446         *
447         * When the bind address is not specified, each bind gets a
448         * new port number, and (for IP transports) we should set
449         * the exclusive flag after every IP bind. That's the
450         * strcmp nc_proto part of the expression below.
451         *
452         * When the bind address IS specified, we need to set the
453         * exclusive flag only after we've bound both IPv6+IPv4,
454         * or the IPv4 bind will fail. Setting the exclusive flag
455         * after the "tcp" or "udp" transport bind does that.
456         * That's the strcmp nc_netid part below.
457         */
458 if (nconf != NULL && ((bindaddr == NULL &&
459     (strcmp(nconf->nc_proto, NC_TCP) == 0 ||
460     strcmp(nconf->nc_proto, NC_UDP) == 0)) ||
461     (strcmp(nconf->nc_netid, "tcp") == 0 ||
462     strcmp(nconf->nc_netid, "udp") == 0))) {
463     bool_t exclbind = FALSE;
464     (void) rpc_control(_RPC_SVC_EXCLBIND_GET, &exclbind);
465     if (exclbind &&
466         _rpc_tli_set_options(fd, SOL_SOCKET,
467             SO_EXCLBIND, 1) < 0) {
468         syslog(LOG_ERR,
469             "svc_tli_create: can't set EXCLBIND [netid='%s']",
470             nconf->nc_netid);
471         goto freedata;
472     }
473 }
474
475 /* Enable options of returning the ip's for udp */
476 if (nconf) {
477     int ret = 0;
478     if (strcmp(nconf->nc_netid, "udp6") == 0) {
479         ret = _rpc_tli_set_options(fd, IPPROTO_IPV6,
480             IPV6_RECVPKTINFO, 1);
481     if (ret < 0) {
482         char errorstr[100];
483
484         __tli_sys_strerror(errorstr,
485             sizeof(errorstr), t_errno, errno);
486         (void) syslog(LOG_ERR,
487             "svc_tli_create: "
488             "IPV6_RECVPKTINFO(2): %s",
489             errorstr);
490         goto freedata;
491     }
492     } else if (strcmp(nconf->nc_netid, "udp") == 0) {
493         ret = _rpc_tli_set_options(fd, IPPROTO_IP,
494             IP_RECVSTADDR, 1);
495     if (ret < 0) {
496         char errorstr[100];
497
498         __tli_sys_strerror(errorstr,
499             sizeof(errorstr), t_errno, errno);
500         (void) syslog(LOG_ERR,
501             "svc_tli_create: "
502             "IP_RECVSTADDR(2): %s", errorstr);
503         goto freedata;
504     }
}

```

```

505
506
507         }
508         break;
509
510     case T_IDLE:
511         if (bindaddr) {
512             /* Copy the entire stuff in tres */
513             if (tres->addr maxlen < bindaddr->addr.len) {
514                 (void) syslog(LOG_ERR,
515                     "svc_tli_create: illegal netbuf length");
516             }
517             tres->addr.len = bindaddr->addr.len;
518             (void) memcpy(tres->addr.buf, bindaddr->addr.buf,
519                     (int) tres->addr.len);
520         } else
521             if (t_getname(fd, &(tres->addr), LOCALNAME) == -1)
522                 tres->addr.len = 0;
523
524         break;
525     case T_INREL:
526         (void) t_rcvrel(fd);
527         (void) t_sndrel(fd);
528         (void) syslog(LOG_ERR, "svc_tli_create: other side wants to "
529                     "release connection");
530         goto freedata;
531
532     case T_INCON:
533         /* Do nothing here. Assume this is handled in rendezvous */
534         break;
535     case T_DATABFER:
536         /*
537          * This takes care of the case where a fd
538          * is passed on which a connection has already
539          * been accepted.
540          */
541         if (t_getname(fd, &(tres->addr), LOCALNAME) == -1)
542             tres->addr.len = 0;
543
544         break;
545     default:
546         (void) syslog(LOG_ERR,
547                     "svc_tli_create: connection in a wierd state (%d)", state);
548         goto freedata;
549
550     /*
551      * call transport specific function.
552      */
553     switch (tinfo.servtype) {
554         case T_COTS_ORD:
555         case T_COTS:
556             if (state == T_DATABFER)
557                 xprt = svc_fd_create_private(fd, sendsz,
558                                              recvsz);
559             else
560                 xprt = svc_vc_create_private(fd, sendsz,
561                                              recvsz);
561             if (!nconf || !xprt)
562                 break;
563             if ((tinfo.servtype == T_COTS_ORD) &&
564                 (state != T_DATABFER) &&
565                 (strcmp(nconf->nc_protomly, "inet") == 0))
566                 (void) __svc_vc_setflag(xprt, TRUE);
567             break;
568     case T_CLTS:
569         xprt = svc_dg_create_private(fd, sendsz, recvsz);
570         break;

```

```

571         default:
572             (void) syslog(LOG_ERR,
573                         "svc_tli_create: bad service type");
574             goto freedata;
575     }
576     if (xprt == NULL)
577     /*
578      * The error messages here are spitted out by the lower layers:
579      * svc_vc_create(), svc_fd_create() and svc_dg_create().
580      */
581     goto freedata;
582
583     /* fill in the other xprt information */
584
585     /* Assign the local bind address */
586     xprt->xp_ltaddr = tres->addr;
587     /* Fill in type of service */
588     xprt->xp_type = tinfo.servtype;
589     tres->addr.buf = NULL;
590     (void) t_free((char *)tres, T_BIND);
591     tres = NULL;
592
593     xprt->xp_rtaddr.len = 0;
594     xprt->xp_rtaddr maxlen = __rpc_get_a_size(tinfo.addr);
595
596     /* Allocate space for the remote bind info */
597     if ((xprt->xp_rtaddr.buf = malloc(xprt->xp_rtaddr maxlen)) == NULL) {
598         (void) syslog(LOG_ERR, "svc_tli_create: No memory!");
599         goto freedata;
600     }
601
602     if (nconf) {
603         xprt->xp_netid = strdup(nconf->nc_netid);
604         if (xprt->xp_netid == NULL) {
605             if (xprt->xp_rtaddr.buf)
606                 free(xprt->xp_rtaddr.buf);
607             syslog(LOG_ERR, "svc_tli_create: strdup failed!");
608             goto freedata;
609         }
610         xprt->xp_tp = strdup(nconf->nc_device);
611         if (xprt->xp_tp == NULL) {
612             if (xprt->xp_rtaddr.buf)
613                 free(xprt->xp_rtaddr.buf);
614             if (xprt->xp_netid)
615                 free(xprt->xp_netid);
616             syslog(LOG_ERR, "svc_tli_create: strdup failed!");
617             goto freedata;
618         }
619     }
620
621     /*
622      * if (madefd && (tinfo.servtype == T_CLTS))
623      *     (void) ioctl(fd, I_POP, NULL);
624     */
625     xprt_register(xprt);
626     return (xprt);
627
628 freedata:
629     if (madefd)
630         (void) t_close(fd);
631     if (tres)
632         (void) t_free((char *)tres, T_BIND);
633     if (xprt) {
634         if (!madefd) /* so that svc_destroy doesn't close fd */
635             xprt->xp_fd = RPC_ANYFD;
636         SVC_DESTROY(xprt);

```

```

637         }
638         return (NULL);
639     }


---


unchanged portion omitted

```

```
new/usr/src/lib/libshare/nfs/libshare_nfs.c
```

```
*****
```

```
77000 Mon Jul 17 16:15:18 2017
```

```
new/usr/src/lib/libshare/nfs/libshare_nfs.c
```

```
7569 statd support to run on a fixed port
```

```
Portions contributed by: Paul Dagnelie <pcd@delphix.com>
```

```
Reviewed by: Evan Layton <evan.layton@nexenta.com>
```

```
Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>
```

```
7577 mountd support to run on a fixed port
```

```
Portions contributed by: Paul Dagnelie <pcd@delphix.com>
```

```
Reviewed by: Evan Layton <evan.layton@nexenta.com>
```

```
Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>
```

```
*****
```

```
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) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
```

```
24 * Copyright 2016 Nexenta Systems, Inc.
```

```
25 * Copyright (c) 2014, 2016 by Delphix. All rights reserved.
```

```
26 */
```

```
28 */
```

```
29 * NFS specific functions
```

```
30 */
```

```
31 #include <stdio.h>
```

```
32 #include <string.h>
```

```
33 #include <ctype.h>
```

```
34 #include <stdlib.h>
```

```
35 #include <unistd.h>
```

```
36 #include <zone.h>
```

```
37 #include <errno.h>
```

```
38 #include <locale.h>
```

```
39 #include <signal.h>
```

```
40 #include <strings.h>
```

```
41 #include "libshare.h"
```

```
42 #include "libshare_impl.h"
```

```
43 #include <nfs/export.h>
```

```
44 #include <pwd.h>
```

```
45 #include <grp.h>
```

```
46 #include <limits.h>
```

```
47 #include <libscf.h>
```

```
48 #include <syslog.h>
```

```
49 #include <rpccsv/daemon_utils.h>
```

```
50 #include "nfslog_config.h"
```

```
51 #include "nfslogtab.h"
```

```
52 #include "libshare_nfs.h"
```

```
53 #include <nfs/nfs.h>
```

```
54 #include <nfs/nfssys.h>
```

```
1
```

```
new/usr/src/lib/libshare/nfs/libshare_nfs.c
```

```
55 #include <netconfig.h>
```

```
56 #include "smfcfg.h"
```

```
58 /* should really be in some global place */
```

```
59 #define DEF_WIN 30000
```

```
60 #define OPT_CHUNK 1024
```

```
62 int debug = 0;
```

```
64 #define NFS_SERVER_SVC "svc:/network/nfs/server:default"
```

```
65 #define NFS_CLIENT_SVC (char *)"svc:/network/nfs/client:default"
```

```
67 /* internal functions */
```

```
68 static int nfs_init();
```

```
69 static void nfs_fini();
```

```
70 static int nfs_enable_share(sa_share_t);
```

```
71 static int nfs_disable_share(sa_share_t, char *);
```

```
72 static int nfs_validate_property(sa_handle_t, sa_property_t, sa_optionset_t);
```

```
73 static int nfs_validate_security_mode(char *);
```

```
74 static int nfs_is_security_opt(char *);
```

```
75 static int nfs_parse_legacy_options(sa_group_t, char *);
```

```
76 static char *nfs_format_options(sa_group_t, int);
```

```
77 static int nfs_set_proto_prop(sa_property_t);
```

```
78 static sa_protocol_properties_t nfs_get_proto_set();
```

```
79 static char *nfs_get_status();
```

```
80 static char *nfs_space_alias(char *);
```

```
81 static uint64_t nfs_features();
```

```
83 /*
```

```
84 * ops vector that provides the protocol specific info and operations
```

```
85 * for share management.
```

```
86 */
```

```
88 struct sa_plugin_ops sa_plugin_ops = {
```

```
89     SA_PLUGIN_VERSION,
```

```
90     "nfs",
```

```
91     nfs_init,
```

```
92     nfs_fini,
```

```
93     nfs_enable_share,
```

```
94     nfs_disable_share,
```

```
95     nfs_validate_property,
```

```
96     nfs_validate_security_mode,
```

```
97     nfs_is_security_opt,
```

```
98     nfs_parse_legacy_options,
```

```
99     nfs_format_options,
```

```
100    nfs_set_proto_prop,
```

```
101    nfs_get_proto_set,
```

```
102    nfs_get_status,
```

```
103    nfs_space_alias,
```

```
104    NULL, /* update_legacy */
```

```
105    NULL, /* delete_legacy */
```

```
106    NULL, /* change_notify */
```

```
107    NULL, /* enable_resource */
```

```
108    NULL, /* disable_resource */
```

```
109    nfs_features,
```

```
110    NULL, /* transient shares */
```

```
111    NULL, /* notify_resource */
```

```
112    NULL, /* rename_resource */
```

```
113    NULL, /* run_command */
```

```
114    NULL, /* command_help */
```

```
115    NULL, /* delete_proto_section */
```

```
116};
```

```
117 unchanged_portion_omitted
```

```
2494 #define PROTO_OPT_NFSD_SERVERS 0
```

```
2495 {"nfsd_servers",
```

```
2496     "servers", PROTO_OPT_NFSD_SERVERS, OPT_TYPE_NUMBER, 1024, SVC_NFSD,
```

```
2
```

```

2497     1, INT32_MAX},
2498 #define PROTO_OPT_LOCKD_LISTEN_BACKLOG           1
2499     {"lockd_listen_backlog",
2500       "lockd_listen_backlog", PROTO_OPT_LOCKD_LISTEN_BACKLOG,
2501       OPT_TYPE_NUMBER, 32, SVC_LOCKD, 32, INT32_MAX},
2502 #define PROTO_OPT_LOCKD_SERVERS                  2
2503     {"lockd_servers",
2504       "lockd_servers", PROTO_OPT_LOCKD_SERVERS, OPT_TYPE_NUMBER, 256,
2505       SVC_LOCKD, 1, INT32_MAX},
2506 #define PROTO_OPT_LOCKD_RETRANSMIT_TIMEOUT        3
2507     {"lockd_retransmit_timeout",
2508       "lockd_retransmit_timeout", PROTO_OPT_LOCKD_RETRANSMIT_TIMEOUT,
2509       OPT_TYPE_NUMBER, 5, SVC_LOCKD, 0, INT32_MAX},
2510 #define PROTO_OPT_GRACE_PERIOD                   4
2511     {"grace_period",
2512       "grace_period", PROTO_OPT_GRACE_PERIOD, OPT_TYPE_NUMBER, 90,
2513       SVC_LOCKD, 0, INT32_MAX},
2514 #define PROTO_OPT_NFS_SERVER_VERSMIN            5
2515     {"nfs_server_versmin",
2516       "server_versmin", PROTO_OPT_NFS_SERVER_VERSMIN, OPT_TYPE_NUMBER,
2517       (int)NFS_VERSMIN_DEFAULT, SVC_NFSD|SVC_MOUNTD, NFS_VERSMIN,
2518       NFS_VERSMAX, "server_versmax", OPT_CMP_LE},
2519 #define PROTO_OPT_NFS_SERVER_VERSMAX            6
2520     {"nfs_server_versmax",
2521       "server_versmax", PROTO_OPT_NFS_SERVER_VERSMAX, OPT_TYPE_NUMBER,
2522       (int)NFS_VERSMAX_DEFAULT, SVC_NFSD|SVC_MOUNTD, NFS_VERSMIN,
2523       NFS_VERSMAX, "server_versmin", OPT_CMP_GE},
2524 #define PROTO_OPT_NFS_CLIENT_VERSMIN            7
2525     {"nfs_client_versmin",
2526       "client_versmin", PROTO_OPT_NFS_CLIENT_VERSMIN, OPT_TYPE_NUMBER,
2527       (int)NFS_VERSMIN_DEFAULT, SVC_CLIENT, NFS_VERSMIN, NFS_VERSMAX,
2528       "client_versmax", OPT_CMP_LE},
2529 #define PROTO_OPT_NFS_CLIENT_VERSMAX            8
2530     {"nfs_client_versmax",
2531       "client_versmax", PROTO_OPT_NFS_CLIENT_VERSMAX, OPT_TYPE_NUMBER,
2532       (int)NFS_VERSMAX_DEFAULT, SVC_CLIENT, NFS_VERSMIN, NFS_VERSMAX,
2533       "client_versmin", OPT_CMP_GB},
2534 #define PROTO_OPT_NFS_SERVER_DELEGATION         9
2535     {"nfs_server_delegation",
2536       "server_delegation", PROTO_OPT_NFS_SERVER_DELEGATION,
2537       OPT_TYPE_ONOFF, NFS_SERVER_DELEGATION_DEFAULT, SVC_NFSD, 0, 0},
2538 #define PROTO_OPT_NFSMAPID_DOMAIN              10
2539     {"nfsmapid_domain",
2540       "nfsmapid_domain", PROTO_OPT_NFSMAPID_DOMAIN, OPT_TYPE_DOMAIN,
2541       NULL, SVC_NFSMAPID, 0, 0},
2542 #define PROTO_OPT_NFSD_MAX_CONNECTIONS          11
2543     {"nfssd_max_connections",
2544       "max_connections", PROTO_OPT_NFSD_MAX_CONNECTIONS,
2545       OPT_TYPE_NUMBER, -1, SVC_NFSD, -1, INT32_MAX},
2546 #define PROTO_OPT_NFSD_PROTOCOL                 12
2547     {"nfssd_protocol",
2548       "protocol", PROTO_OPT_NFSD_PROTOCOL, OPT_TYPE_PROTOCOL, 0,
2549       SVC_NFSD, 0, 0},
2550 #define PROTO_OPT_NFSD_LISTEN_BACKLOG          13
2551     {"nfssd_listen_backlog",
2552       "listen_backlog", PROTO_OPT_NFSD_LISTEN_BACKLOG,
2553       OPT_TYPE_NUMBER, 0, SVC_NFSD, 0, INT32_MAX},
2554 #define PROTO_OPT_NFSD_DEVICE                  14
2555     {"nfssd_device",
2556       "device", PROTO_OPT_NFSD_DEVICE,
2557       OPT_TYPE_STRING, NULL, SVC_NFSD, 0, 0},
2558 #define PROTO_OPT_MOUNTD_LISTEN_BACKLOG        15
2559     {"mountd_listen_backlog",
2560       "mountd_listen_backlog", PROTO_OPT_MOUNTD_LISTEN_BACKLOG,
2561       OPT_TYPE_NUMBER, 64, SVC_NFSD|SVC_MOUNTD, 1, INT32_MAX},
2562 #define PROTO_OPT_MOUNTD_MAX_THREADS           16

```

```

2563     {"mounted_max_threads",
2564       "mounted_max_threads", PROTO_OPT_MOUNTD_MAX_THREADS,
2565       OPT_TYPE_NUMBER, 16, SVC_NFSD|SVC_MOUNTD, 1, INT32_MAX},
2566 #define PROTO_OPT_MOUNTD_PORT                 17
2567     {"mounted_port",
2568       "mountd_port", PROTO_OPT_MOUNTD_PORT,
2569       OPT_TYPE_NUMBER, 0, SVC_MOUNTD, 1, UINT16_MAX},
2570 #define PROTO_OPT_STATD_PORT                  18
2571     {"statd_port",
2572       "statd_port", PROTO_OPT_STATD_PORT,
2573       OPT_TYPE_NUMBER, 0, SVC_STATD, 1, UINT16_MAX},
2574     {NULL};
2575 };

```

unchanged portion omitted

```
*****
19174 Mon Jul 17 16:15:19 2017
new/usr/src/man/man3nsl/Makefile
8330 Add svc_tp_create_addr to libnsl
Reviewed by: Paul Dagnelie <pcd@delphix.com>
Reviewed by: Evan Layton <evan.layton@nexenta.com>
Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>
*****
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License (" CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet
9 # at http://www.illumos.org/license/CDDL.
10 #

12 #
13 # Copyright 2011, Richard Lowe
14 # Copyright 2017 Nexenta Systems, Inc. All rights reserved.
14 # Copyright 2013 Nexenta Systems, Inc. All rights reserved.
15 #

17 include      $(SRC)/Makefile.master

19 MANSECT=      3nsl

21 MANFILES=      dial.3nsl          \
22                 doconfig.3nsl        \
23                 gethostbyname.3nsl    \
24                 getipsecalgbyname.3nsl \
25                 getipsecpotobyname.3nsl \
26                 getnetconfig.3nsl     \
27                 getnetpath.3nsl      \
28                 getpublickey.3nsl     \
29                 getrpcbyname.3nsl     \
30                 netdir.3nsl          \
31                 nlsgetcall.3nsl       \
32                 nlsprovider.3nsl      \
33                 nlsrequest.3nsl       \
34                 rpc.3nsl            \
35                 rpc_clnt_auth.3nsl     \
36                 rpc_clnt_calls.3nsl    \
37                 rpc_clnt_create.3nsl   \
38                 rpc_control.3nsl      \
39                 rpc_gss_get_error.3nsl  \
40                 rpc_gss_get_mechanisms.3nsl \
41                 rpc_gss_get_principal_name.3nsl \
42                 rpc_gss_getcred.3nsl    \
43                 rpc_gss_max_data_length.3nsl \
44                 rpc_gss_mech_to_oid.3nsl \
45                 rpc_gss_seccreate.3nsl   \
46                 rpc_gss_set_callback.3nsl \
47                 rpc_gss_set_defaults.3nsl \
48                 rpc_gss_set_svc_name.3nsl \
49                 rpc_soc.3nsl          \
50                 rpc_svc_calls.3nsl     \
51                 rpc_svc_create.3nsl    \
52                 rpc_svc_err.3nsl       \
53                 rpc_svc_input.3nsl     \
54                 rpc_svc_reg.3nsl       \
55                 rpc_xdr.3nsl          \
56                 rpcbind.3nsl          \
57                 rpcsec_gss.3nsl        \

```

```
58                 secure_rpc.3nsl      \
59                 t_accept.3nsl         \
60                 t_alloc.3nsl          \
61                 t_bind.3nsl          \
62                 t_close.3nsl          \
63                 t_connect.3nsl        \
64                 t_errno.3nsl          \
65                 t_error.3nsl          \
66                 t_free.3nsl          \
67                 t_getinfo.3nsl        \
68                 t_getprotaddr.3nsl     \
69                 t_getstate.3nsl        \
70                 t_listen.3nsl          \
71                 t_look.3nsl          \
72                 t_open.3nsl          \
73                 t_optmgmt.3nsl        \
74                 t_rcv.3nsl          \
75                 t_rcvconnect.3nsl      \
76                 t_rcvdis.3nsl          \
77                 t_rcvrel.3nsl          \
78                 t_rcvreldata.3nsl      \
79                 t_rcvudata.3nsl        \
80                 t_rcvuderr.3nsl        \
81                 t_rcvv.3nsl          \
82                 t_rcvvudata.3nsl      \
83                 t_snd.3nsl          \
84                 t_snddis.3nsl          \
85                 t_sndrel.3nsl          \
86                 t_sndreldata.3nsl      \
87                 t_sndudata.3nsl        \
88                 t_sndv.3nsl          \
89                 t_sndvudata.3nsl      \
90                 t_strerror.3nsl        \
91                 t_sync.3nsl          \
92                 t_sysconf.3nsl        \
93                 t_unbind.3nsl          \
94                 xdri.3nsl          \
95                 xdri_admin.3nsl      \
96                 xdri_complex.3nsl     \
97                 xdri_create.3nsl      \
98                 xdri_simple.3nsl      \
99                 yp_update.3nsl        \
100                ypclnt.3nsl          \
102 MANLINKS=      auth_destroy.3nsl      \
103                authdes_create.3nsl    \
104                authdes_getucred.3nsl  \
105                authdes_seccreate.3nsl \
106                authhone_create.3nsl   \
107                authsys_create.3nsl    \
108                authsys_create_default.3nsl \
109                authunix_create.3nsl   \
110                authunix_create_default.3nsl \
111                callrpc.3nsl          \
112                clnt_broadcast.3nsl    \
113                clnt_call.3nsl          \
114                clnt_control.3nsl      \
115                clnt_create.3nsl        \
116                clnt_create_timed.3nsl  \
117                clnt_create_vers.3nsl    \
118                clnt_create_vers_timed.3nsl \
119                clnt_destroy.3nsl      \
120                clnt_dg_create.3nsl      \
121                clnt_door_create.3nsl   \
122                clnt_freeres.3nsl      \
123                clnt_geterr.3nsl        \

```

```

124      clnt_pcreateerror.3nsl
125      clnt_perrno.3nsl
126      clnt_perror.3nsl
127      clnt_raw_create.3nsl
128      clnt_send.3nsl
129      clnt_spcreateerror.3nsl
130      clnt_sperrno.3nsl
131      clnt_sperror.3nsl
132      clnt_tli_create.3nsl
133      clnt_tp_create.3nsl
134      clnt_tp_create_timed.3nsl
135      clnt_vc_create.3nsl
136      clntraw_create.3nsl
137      clnttcp_create.3nsl
138      clntudp_bufcreate.3nsl
139      clntudp_create.3nsl
140      endhostent.3nsl
141      endnetconfig.3nsl
142      endnetpath.3nsl
143      endrpccent.3nsl
144      freeipsecalgent.3nsl
145      freenetconfigent.3nsl
146      get_myaddress.3nsl
147      gethostbyaddr.3nsl
148      gethostbyaddr_r.3nsl
149      gethostbyname_r.3nsl
150      gethostent.3nsl
151      gethostent_r.3nsl
152      getipscalegbynum.3nsl
153      getipsecpbynum.3nsl
154      getnetconfigent.3nsl
155      getnetname.3nsl
156      getrpcbyname_r.3nsl
157      getrpcbynumber.3nsl
158      getrpcbynumber_r.3nsl
159      getrpcent.3nsl
160      getrpcent_r.3nsl
161      getrpcport.3nsl
162      getsecretkey.3nsl
163      host2netname.3nsl
164      key_decryptsession.3nsl
165      key_encryptsession.3nsl
166      key_gendes.3nsl
167      key_secretkey_is_set.3nsl
168      key_setsecret.3nsl
169      nc_perror.3nsl
170      nc_s perror.3nsl
171      netdir_free.3nsl
172      netdir_getbyaddr.3nsl
173      netdir_getbyname.3nsl
174      netdir_mergeaddr.3nsl
175      netdir_options.3nsl
176      netdir_perror.3nsl
177      netdir_sperror.3nsl
178      netname2host.3nsl
179      netname2user.3nsl
180      pmap_getmaps.3nsl
181      pmap_getport.3nsl
182      pmap_rmtcall.3nsl
183      pmap_set.3nsl
184      pmap_unset.3nsl
185      publickey.3nsl
186      registerrpc.3nsl
187      rpc_broadcast.3nsl
188      rpc_broadcast_exp.3nsl
189      rpc_call.3nsl

```

```

190      rpc_createerr.3nsl
191      rpc_gss_get_mech_info.3nsl
192      rpc_gss_get_versions.3nsl
193      rpc_gss_is_installed.3nsl
194      rpc_gss_qop_to_num.3nsl
195      rpc_gss_svc_max_data_length.3nsl
196      rpc_reg.3nsl
197      rpcb_getaddr.3nsl
198      rpcb_getmaps.3nsl
199      rpcb_gettime.3nsl
200      rpcb_rmtcall.3nsl
201      rpcb_set.3nsl
202      rpcb_unset.3nsl
203      sethostent.3nsl
204      setnetconfig.3nsl
205      setnetpath.3nsl
206      setrpcent.3nsl
207      svc_add_input.3nsl
208      svc_auth_reg.3nsl
209      svc_control.3nsl
210      svc_create.3nsl
211      svc_destroy.3nsl
212      svc_dg_create.3nsl
213      svc_dg_enablecache.3nsl
214      svc_done.3nsl
215      svc_door_create.3nsl
216      svc_exit.3nsl
217      svc_fd_create.3nsl
218      svc_fd_negotiate_ucred.3nsl
219      svc_fds.3nsl
220      svc_fdset.3nsl
221      svc_freeargs.3nsl
222      svc_getargs.3nsl
223      svc_getcaller.3nsl
224      svc_getcallerucred.3nsl
225      svc_getreq.3nsl
226      svc_getreq_common.3nsl
227      svc_getreq_poll.3nsl
228      svc_getreqset.3nsl
229      svc_getrpccaller.3nsl
230      svc_max_pollfd.3nsl
231      svc_pollfd.3nsl
232      svc_raw_create.3nsl
233      svc_reg.3nsl
234      svc_register.3nsl
235      svc_remove_input.3nsl
236      svc_run.3nsl
237      svc_sendreply.3nsl
238      svc_tli_create.3nsl
239      svc_tp_create.3nsl
240      svc_tp_create_addr.3nsl
241      svc_unreg.3nsl
242      svc_unregister.3nsl
243      svc_vc_create.3nsl
244      svccerr_auth.3nsl
245      svccerr_decode.3nsl
246      svccerr_noproc.3nsl
247      svccerr_noprog.3nsl
248      svccerr_progvers.3nsl
249      svccerr_systemerr.3nsl
250      svccerr_weakauth.3nsl
251      svcfd_create.3nsl
252      svccraw_create.3nsl
253      svctcp_create.3nsl
254      svcudp_bufcreate.3nsl
255      svcudp_create.3nsl

```

256 taddr2uaddr.3nsl
257 uaddr2taddr.3nsl
258 undial.3nsl
259 user2netname.3nsl
260 xdr_accepted_reply.3nsl
261 xdr_array.3nsl
262 xdr_authsys_parms.3nsl
263 xdr_authunix_parms.3nsl
264 xdr_bool.3nsl
265 xdr_bytes.3nsl
266 xdr_callhdr.3nsl
267 xdr_callmsg.3nsl
268 xdr_char.3nsl
269 xdr_control.3nsl
270 xdr_destroy.3nsl
271 xdr_double.3nsl
272 xdr_enum.3nsl
273 xdr_float.3nsl
274 xdr_free.3nsl
275 xdr_getpos.3nsl
276 xdr_hyper.3nsl
277 xdr_inline.3nsl
278 xdr_int.3nsl
279 xdr_long.3nsl
280 xdr_longlong_t.3nsl
281 xdr_opaque.3nsl
282 xdr_opaque_auth.3nsl
283 xdr_pointer.3nsl
284 xdr_quadruple.3nsl
285 xdr_reference.3nsl
286 xdr_rejected_reply.3nsl
287 xdr_replaymsg.3nsl
288 xdr_setpos.3nsl
289 xdr_short.3nsl
290 xdr_sizeof.3nsl
291 xdr_string.3nsl
292 xdr_u_char.3nsl
293 xdr_u_hyper.3nsl
294 xdr_u_int.3nsl
295 xdr_u_long.3nsl
296 xdr_u_longlong_t.3nsl
297 xdr_u_short.3nsl
298 xdr_union.3nsl
299 xdr_vector.3nsl
300 xdr_void.3nsl
301 xdr_wrapstring.3nsl
302 xdrcmem_create.3nsl
303 xdrec_create.3nsl
304 xdrec_endofrecord.3nsl
305 xdrec_eof.3nsl
306 xdrec_readbytes.3nsl
307 xdrec_skiprecord.3nsl
308 xdrstdio_create.3nsl
309 xprt_register.3nsl
310 xprt_unregister.3nsl
311 yp_all.3nsl
312 yp_bind.3nsl
313 yp_first.3nsl
314 yp_get_default_domain.3nsl
315 yp_master.3nsl
316 yp_match.3nsl
317 yp_next.3nsl
318 yp_order.3nsl
319 yp_unbind.3nsl
320 yperr_string.3nsl
321 ypprot_err.3nsl

```

388 clnt_create_timed.3ns1
389 clnt_create_vers.3ns1
390 clnt_create_vers_timed.3ns1
391 clnt_destroy.3ns1
392 clnt_dg_create.3ns1
393 clnt_door_create.3ns1
394 clnt_pcreateerror.3ns1
395 clnt_raw_create.3ns1
396 clnt_spcreateerror.3ns1
397 clnt_tli_create.3ns1
398 clnt_tp_create.3ns1
399 clnt_tp_create_timed.3ns1
400 clnt_vc_create.3ns1
401 rpc_createerr.3ns1

403 rpc_gss_get_mech_info.3ns1
404 rpc_gss_get_versions.3ns1
405 rpc_gss_is_installed.3ns1
407 rpc_gss_svc_max_data_length.3ns1
409 rpc_gss_qop_to_num.3ns1

411 authdes_create.3ns1
412 authunix_create.3ns1
413 authunix_create_default.3ns1
414 callrpc.3ns1
415 clnt_broadcast.3ns1
416 clntraw_create.3ns1
417 clnttcp_create.3ns1
418 clntudp_bufcreate.3ns1
419 clntudp_create.3ns1
420 get_myaddress.3ns1
421 getrpport.3ns1
422 pmap_getmaps.3ns1
423 pmap_getport.3ns1
424 pmap_rmtcall.3ns1
425 pmap_set.3ns1
426 pmap_unset.3ns1
427 registerrpc.3ns1
428 svc_fds.3ns1
429 svc_getcaller.3ns1
430 svc_getreq.3ns1
431 svc_register.3ns1
432 svc_unregister.3ns1
433 svcfd_create.3ns1
434 svccraw_create.3ns1
435 svctcp_create.3ns1
436 svcudp_bufcreate.3ns1
437 svcudp_create.3ns1
438 xdr_authunix_parms.3ns1

440 svc_dg_enablecache.3ns1
441 svc_done.3ns1
442 svc_exit.3ns1
443 svc_fd_negotiate_ucred.3ns1
444 svc_fdset.3ns1
445 svc_freeargs.3ns1
446 svc_getargs.3ns1
447 svc_getcallerucred.3ns1
448 svc_getreq_common.3ns1
449 svc_getreq_poll.3ns1
450 svc_getreqset.3ns1
451 svc_getrppcaller.3ns1
452 svc_max_pollfd.3ns1
453 svc_pollfd.3ns1

```

```

454 svc_run.3ns1
455 svc_sendreply.3ns1
457 svc_control.3ns1
458 svc_create.3ns1
459 svc_destroy.3ns1
460 svc_dg_create.3ns1
461 svc_door_create.3ns1
462 svc_fd_create.3ns1
463 svc_raw_create.3ns1
464 svc_tli_create.3ns1
465 svc_tp_create.3ns1
466 svc_tp_create_addr.3ns1
467 svc_vc_create.3ns1

469 svcerr_auth.3ns1
470 svcerr_decode.3ns1
471 svcerr_noproc.3ns1
472 svcerr_noprog.3ns1
473 svcerr_progvers.3ns1
474 svcerr_systemerr.3ns1
475 svcerr_weakauth.3ns1

477 svc_add_input.3ns1
478 svc_remove_input.3ns1
480 rpc_reg.3ns1
481 svc_auth_reg.3ns1
482 svc_reg.3ns1
483 svc_unreg.3ns1
484 xprt_register.3ns1
485 xprt_unregister.3ns1
487 xdr_accepted_reply.3ns1
488 xdr_authsys_parms.3ns1
489 xdr_callhdr.3ns1
490 xdr_calmmsg.3ns1
491 xdr_opaque_auth.3ns1
492 xdr_rejected_reply.3ns1
493 xdr_replaymsg.3ns1
495 rpcb_getaddr.3ns1
496 rpcb_getmaps.3ns1
497 rpcb_gettime.3ns1
498 rpcb_rmtcall.3ns1
499 rpcb_set.3ns1
500 rpcb_unset.3ns1
502 authdes_getucred.3ns1
503 authdes_seccreate.3ns1
504 getnetname.3ns1
505 host2netname.3ns1
506 key_decryptsession.3ns1
507 key_encryptsession.3ns1
508 key_gendes.3ns1
509 key_secretkey_is_set.3ns1
510 key_setsecret.3ns1
511 netname2host.3ns1
512 netname2user.3ns1
513 user2netname.3ns1
515 xdr_control.3ns1
516 xdr_getpos.3ns1
517 xdr_inline.3ns1
518 xdr_setpos.3ns1
519 xdr_sizeof.3ns1

```

```
520 xdrrec_endofrecord.3nsl      := LINKSRC = xdr_admin.3nsl
521 xdrrec_eof.3nsl              := LINKSRC = xdr_admin.3nsl
522 xdrrec_readbytes.3nsl        := LINKSRC = xdr_admin.3nsl
523 xdrrec_skiprecord.3nsl       := LINKSRC = xdr_admin.3nsl

525 xdr_array.3nsl              := LINKSRC = xdr_complex.3nsl
526 xdr_bytes.3nsl               := LINKSRC = xdr_complex.3nsl
527 xdr_opaque.3nsl              := LINKSRC = xdr_complex.3nsl
528 xdr_pointer.3nsl             := LINKSRC = xdr_complex.3nsl
529 xdr_reference.3nsl            := LINKSRC = xdr_complex.3nsl
530 xdr_string.3nsl              := LINKSRC = xdr_complex.3nsl
531 xdr_union.3nsl               := LINKSRC = xdr_complex.3nsl
532 xdr_vector.3nsl              := LINKSRC = xdr_complex.3nsl
533 xdr_wrapstring.3nsl           := LINKSRC = xdr_complex.3nsl

535 xdr_destroy.3nsl              := LINKSRC = xdr_create.3nsl
536 xdrmem_create.3nsl             := LINKSRC = xdr_create.3nsl
537 xdrrec_create.3nsl             := LINKSRC = xdr_create.3nsl
538 xdrstdio_create.3nsl           := LINKSRC = xdr_create.3nsl

540 xdr_bool.3nsl                := LINKSRC = xdr_simple.3nsl
541 xdr_char.3nsl                 := LINKSRC = xdr_simple.3nsl
542 xdr_double.3nsl               := LINKSRC = xdr_simple.3nsl
543 xdr_enum.3nsl                 := LINKSRC = xdr_simple.3nsl
544 xdr_float.3nsl                := LINKSRC = xdr_simple.3nsl
545 xdr_free.3nsl                  := LINKSRC = xdr_simple.3nsl
546 xdr_hyper.3nsl                := LINKSRC = xdr_simple.3nsl
547 xdr_int.3nsl                  := LINKSRC = xdr_simple.3nsl
548 xdr_long.3nsl                 := LINKSRC = xdr_simple.3nsl
549 xdr_longlong_t.3nsl             := LINKSRC = xdr_simple.3nsl
550 xdr_quadruple.3nsl             := LINKSRC = xdr_simple.3nsl
551 xdr_short.3nsl                 := LINKSRC = xdr_simple.3nsl
552 xdr_u_char.3nsl                := LINKSRC = xdr_simple.3nsl
553 xdr_u_hyper.3nsl               := LINKSRC = xdr_simple.3nsl
554 xdr_u_int.3nsl                 := LINKSRC = xdr_simple.3nsl
555 xdr_u_long.3nsl                := LINKSRC = xdr_simple.3nsl
556 xdr_u_longlong_t.3nsl            := LINKSRC = xdr_simple.3nsl
557 xdr_u_short.3nsl               := LINKSRC = xdr_simple.3nsl
558 xdr_void.3nsl                  := LINKSRC = xdr_simple.3nsl

560 yp_all.3nsl                  := LINKSRC = ypclnt.3nsl
561 yp_bind.3nsl                  := LINKSRC = ypclnt.3nsl
562 yp_first.3nsl                 := LINKSRC = ypclnt.3nsl
563 yp_get_default_domain.3nsl      := LINKSRC = ypclnt.3nsl
564 yp_master.3nsl                 := LINKSRC = ypclnt.3nsl
565 yp_match.3nsl                  := LINKSRC = ypclnt.3nsl
566 yp_next.3nsl                   := LINKSRC = ypclnt.3nsl
567 yp_order.3nsl                  := LINKSRC = ypclnt.3nsl
568 yp_unbind.3nsl                 := LINKSRC = ypclnt.3nsl
569 yperr_string.3nsl               := LINKSRC = ypclnt.3nsl
570 ypprot_err.3nsl                 := LINKSRC = ypclnt.3nsl

572 .KEEP_STATE:

574 include          $(SRC)/man/Makefile.man
576 install:         $(ROOTMANFILES) $(ROOTMANLINKS)
```

```
new/usr/src/man/man3ns1/rpc_svc_create.3ns1
```

```
*****
```

```
14979 Mon Jul 17 16:15:19 2017
```

```
new/usr/src/man/man3ns1/rpc_svc_create.3ns1
```

```
8330 Add svc_tp_create_addr to libns1
```

```
Reviewed by: Paul Dagnelie <pcd@delphix.com>
```

```
Reviewed by: Evan Layton <evan.layton@nexenta.com>
```

```
Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>
```

```
*****
```

```
1 '\\" te
```

```
2 '\\" Copyright 2017 Nexenta Systems, Inc. All Rights Reserved.
```

```
2 '\\" Copyright 2014 Nexenta Systems, Inc. All Rights Reserved.
```

```
3 '\\" Copyright 1989 AT&T
```

```
4 '\\" Copyright (C) 2005, Sun Microsystems, Inc. All Rights Reserved.
```

```
5 '\\" The contents of this file are subject to the terms of the Common Development
```

```
6 '\\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
```

```
7 '\\" When distributing Covered Code, include this CDDL HEADER in each file and in
```

```
8 .TH RPC_SVC_CREATE 3NSL "Jun 19, 2017"
```

```
8 .TH RPC_SVC_CREATE 3NSL "May 18, 2017"
```

```
9 .SH NAME
```

```
10 rpc_svc_create, svc_control, svc_create, svc_destroy, svc_dg_create,
```

```
11 svc_fd_create, svc_raw_create, svc_tli_create, svc_tp_create,
```

```
12 svc_tp_create_addr, svc_vc_create,
```

```
11 svc_fd_create, svc_raw_create, svc_tli_create, svc_tp_create, svc_vc_create,
```

```
13 svc_door_create \- server handle creation routines
```

```
14 .SH SYNOPSIS
```

```
15 .LP
```

```
16 .nf
```

```
17 #include <rpc/rpc.h>
```

```
19 \fBbool_t\fR \fBsvc_control\fR(\fBSVCXPRT *\fR\fIsvc\fR, \fBconst uint_t\fR \fI
```

```
20 .fi
```

```
22 .LP
```

```
23 .nf
```

```
24 \fBint\fR \fBsvc_create\fR(\fBconst void (*\fR\fIdispatch\fR)(const struct svc_r
```

```
25     const SVCXPRT *), \fBconst rpcprog_t\fR \fIpnum\fR, \fBconst rpcvers_t\fR
```

```
26     \fBconst char *\fR\fInettype\fR);
```

```
27 .fi
```

```
29 .LP
```

```
30 .nf
```

```
31 \fBvoid\fR \fBsvc_destroy\fR(\fBSVCXPRT *\fR\fIxprt\fR);
```

```
32 .fi
```

```
34 .LP
```

```
35 .nf
```

```
36 \fBSVCXPRT *\fR\fBsvc_dg_create\fR(\fBconst int\fR \fIfildes\fR, \fBconst uint_t
```

```
37     \fBconst uint_t\fR \fIrecvsz\fR);
```

```
38 .fi
```

```
40 .LP
```

```
41 .nf
```

```
42 \fBSVCXPRT *\fR\fBsvc_fd_create\fR(\fBconst int\fR \fIfildes\fR, \fBconst uint_t
```

```
43     \fBconst uint_t\fR \fIrecvsz\fR);
```

```
44 .fi
```

```
46 .LP
```

```
47 .nf
```

```
48 \fBSVCXPRT *\fR\fBsvc_raw_create\fR(void);
```

```
49 .fi
```

```
51 .LP
```

```
52 .nf
```

```
53 \fBSVCXPRT *\fR\fBsvc_tli_create\fR(\fBconst int\fR \fIfildes\fR, \fBconst struc
```

```
54     \fBconst struct t_bind *\fR\fBind_info\fR, \fBconst uint_t\fR \fIsendsz\fR
```

```
53     \fBconst struct t_bind *\fR\fBind_addr\fR, \fBconst uint_t\fR \fIsendsz\fR
```

```
1
```

```
new/usr/src/man/man3ns1/rpc_svc_create.3ns1
```

```
55     \fBconst uint_t\fR \fIrecvsz\fR);
```

```
56 .fi
```

```
58 .LP
```

```
59 .nf
```

```
60 \fBSVCXPRT *\fR\fBsvc_tp_create\fR(\fBconst void (*\fR\fIdispatch\fR)
```

```
61     (const struct svc_req *, const SVCXPRT *), \fBconst rpcprog_t\fR \fIpnum
```

```
62     \fBconst rpcvers_t\fR \fIversnum\fR, \fBconst struct netconfig *\fR\fInetco
```

```
63 .fi
```

```
65 .LP
```

```
66 .nf
```

```
67 \fBSVCXPRT *\fR\fBsvc_tp_create_addr\fR(\fBconst void (*\fR\fIdispatch\fR)
```

```
68     (const struct svc_req *, const SVCXPRT *), \fBconst rpcprog_t\fR \fIpnum
```

```
69     \fBconst rpcvers_t\fR \fIversnum\fR, \fBconst struct netconfig *\fR\fInetco
```

```
70 .fi
```

```
71 );
```

```
72 .fi
```

```
74 .LP
```

```
75 .nf
```

```
76 \fBSVCXPRT *\fR\fBsvc_vc_create\fR(\fBconst int\fR \fIfildes\fR, \fBconst uint_t
```

```
77     \fBconst uint_t\fR \fIrecvsz\fR);
```

```
78 .fi
```

```
80 .LP
```

```
81 .nf
```

```
82 \fBSVCXPRT *\fR\fBsvc_door_create\fR(\fBvoid (*\fR\fIdispatch\fR)(struct svc_req
```

```
83     \fBconst rpcprog_t\fR \fIpnum\fR, \fBconst rpcvers_t\fR \fIversnum\fR,
```

```
84     \fBconst uint_t\fR \fIsendsz\fR);
```

```
85 .fi
```

```
87 .SH DESCRIPTION
```

```
88 .LP
```

```
89 These routines are part of the \fBRPC\fR library which allows C language  
90 programs to make procedure calls on servers across the network. These routines  
91 deal with the creation of service handles. Once the handle is created, the  
92 server can be invoked by calling \fBsvc_run()\fR.
```

```
93 .SS "Routines"
```

```
94 .LP
```

```
95 See \fBrcp\fR(3NSL) for the definition of the \fBSVCXPRT\fR data structure.
```

```
96 .sp
```

```
97 .ne 2
```

```
98 .na
```

```
99 \fB\Bsvc_control()\fR\fR
```

```
100 .ad
```

```
101 .RS 15n
```

```
91 .RS 21n
```

```
102 A function to change or retrieve information about a service object. \fIreq\fR  
103 indicates the type of operation and \fIinfo\fR is a pointer to the information.  
104 The supported values of \fIreq\fR, their argument types, and what they do are:
```

```
105 .sp
```

```
106 .ne 2
```

```
107 .na
```

```
108 \fB\Bsvc_get_versquiet\fR\fR
```

```
109 .ad
```

```
110 .RS 10n
```

```
100 .RS 25n
```

```
111 If a request is received for a program number served by this server but the  
112 version number is outside the range registered with the server, an  
113 \fBPROGVERSMISMATCH\fR error will normally be returned. \fIinfo\fR should  
114 be a pointer to an integer. Upon successful completion of the
```

```
115 \fBsvc_get_versquiet\fR request, *\fIinfo\fR contains an integer which  
116 describes the server's current behavior: \fB0\fR indicates normal server  
117 behavior, that is, an \fBPROGVERSMISMATCH\fR error will be returned.  
118 \fB1\fR indicates that the out of range request will be silently ignored.
```

```
2
```

```

119 .RE
121 .sp
122 .ne 2
123 .na
124 \fB\fBSVCSET_VERSQUIET\fR\fR
125 .ad
126 .RS 10n
127 If a request is received for a program number served by this server but the
128 version number is outside the range registered with the server, an
129 \fBRPC_PROGVERSMISMATCH\fR error will normally be returned. It is sometimes
130 desirable to change this behavior. \fIinfo\fR should be a pointer to an integer
131 which is either \fb0\fR, indicating normal server behavior and an
132 \fBRPC_PROGVERSMISMATCH\fR error will be returned, or \fb1\fR, indicating that
133 the out of range request should be silently ignored.
134 .RE

136 .sp
137 .ne 2
138 .na
139 \fB\fBSVCGET_XID\fR\fR
140 .ad
141 .RS 10n
142 Returns the transaction \fBID\fR of connection\(\mioriented and connectionless
143 transport service calls. The transaction \fBID\fR assists in uniquely
144 identifying client requests for a given \fBRPC\fR version, program number,
145 procedure, and client. The transaction \fBID\fR is extracted from the service
146 transport handle \fIsvc\fR. \fIinfo\fR must be a pointer to an unsigned long.
147 Upon successful completion of the \fBSVCGET_XID\fR request, *\fIinfo\fR
148 contains the transaction \fBID\fR. Note that rendezvous and raw service
149 handles do not define a transaction \fBID\fR. Thus, if the service handle is
150 of rendezvous or raw type, and the request is of type \fBSVCGET_XID\fR
151 \fBsvc_control()\fR will return \fBFALSE\fR. Note also that the transaction
152 \fBID\fR read by the server can be set by the client through the suboption
153 \fBCLSET_XID\fR in \fBclnt_control()\fR. See \fBclnt_create\fR(3NSL)
154 .RE

156 .sp
157 .ne 2
158 .na
159 \fB\fBSVCSET_RECVERRHANDLER\fR\fR
160 .ad
161 .RS 10n
162 Attaches or detaches a disconnection handler to the service handle, \fIsvc\fR,
163 that will be called when a transport error arrives during the reception of a
164 request or when the server is waiting for a request and the connection shuts
165 down. This handler is only useful for a connection oriented service handle.
166 .sp
167 \fIinfo\fR contains the address of the error handler to attach, or \fINULL\fR
168 to detach a previously defined one. The error handler has two arguments. It has
169 a pointer to the erroneous service handle. It also has an integer that
170 indicates if the full service is closed (when equal to zero), or that only one
171 connection on this service is closed (when not equal to zero).
172 .sp
173 .in +2
174 .nf
175 void handler (const SVCXPRT *svc, const bool_t isAConnection);
176 .fi
177 .in -2

179 With the service handle address, \fIsvc\fR, the error handler is able to detect
180 which connection has failed and to begin an error recovery process. The error
181 handler can be called by multiple threads and should be implemented in an

```

```

182 MT-safe way.
183 .RE
185 .sp
186 .ne 2
187 .na
188 \fB\fBSVCGET_RECVERRHANDLER\fR\fR
189 .ad
190 .RS 10n
191 .RS 25n
192 Upon successful completion of the \fBSVCGET_RECVERRHANDLER\fR request,
193 \fIinfo\fR contains the address of the handler for receiving errors. Upon
194 failure, \fIinfo\fR contains \fINULL\fR.
194 .RE

196 .sp
197 .ne 2
198 .na
199 \fB\fBSVCSET_CONNMAXREC\fR\fR
200 .ad
201 .RS 10n
202 Set the maximum record size (in bytes) and enable non-blocking mode for this
203 service handle. Value can be set and read for both connection and
204 non-connection oriented transports, but is silently ignored for the
205 non-connection oriented case. The \fIinfo\fR argument should be a pointer to an
206 \fBint\fR.
207 .RE

209 .sp
210 .ne 2
211 .na
212 \fB\fBSVCGET_CONNMAXREC\fR\fR
213 .ad
214 .RS 10n
204 .RS 25n
215 Get the maximum record size for this service handle. Zero means no maximum in
216 effect and the connection is in blocking mode. The result is not significant
217 for non-connection oriented transports. The \fIinfo\fR argument should be a
218 pointer to an \fBint\fR.
219 .RE

221 This routine returns TRUE if the operation was successful. Otherwise, it
222 returns false.
223 .RE

225 .sp
226 .ne 2
227 .na
228 \fB\fBsvc_create()\fR\fR
229 .ad
230 .RS 15n
220 .RS 21n
231 \fBsvc_create()\fR creates server handles for all the transports belonging to
232 the class \fInettype\fR.
233 .sp
234 \fInettype\fR defines a class of transports which can be used for a particular
235 application. The transports are tried in left to right order in \fBNETPATH\fR
236 variable or in top to bottom order in the netconfig database. If \fInettype\fR
237 is \fINULL\fR it defaults to \fBnetpath\fR.
238 .sp
239 \fBsvc_create()\fR registers itself with the \fBrpcbind\fR service (see
240 \fBrpcbind\fR(1M)). \fIdispatch\fR is called when there is a remote procedure
241 call for the given \fIprognam\fR and \fIversnum\fR; this requires calling
242 \fBsvc_run()\fR (see \fBsvc_run()\fR in \fBrpc_svc_calls\fR(3NSL)). If
243 \fBsvc_create()\fR succeeds, it returns the number of server handles it

```

```

244 created, otherwise it returns \fB0\fR and an error message is logged.
245 .RE

247 .sp
248 .ne 2
249 .na
250 \fB\fBsvc_destroy()\fR\fR
251 .ad
252 .RS 15n
253 .RS 21n
253 A function macro that destroys the \fBRPC\fR service handle \fIxprt\fR.
254 Destruction usually involves deallocation of private data structures, including
255 \fIxprt\fR itself. Use of \fIxprt\fR is undefined after calling this routine.
256 .RE

258 .sp
259 .ne 2
260 .na
261 \fB\fBsvc_dg_create()\fR\fR
262 .ad
263 .RS 15n
263 .RS 21n
264 This routine creates a connectionless \fBRPC\fR service handle, and returns a
265 pointer to it. This routine returns \fINULL\fR if it fails, and an error
266 message is logged. \fIsendsz\fR and \fIrecvsz\fR are parameters used to specify
267 the size of the buffers. If they are \fB0\fR, suitable defaults are chosen. The
268 file descriptor \fifildes\fR should be open and bound. The server is not
269 registered with \fBrpcbind\fR(1M).
270 .sp
271 Warning: since connectionless-based \fBRPC\fR messages can only hold limited
272 amount of encoded data, this transport cannot be used for procedures that take
273 large arguments or return huge results.
274 .RE

276 .sp
277 .ne 2
278 .na
279 \fB\fBsvc_fd_create()\fR\fR
280 .ad
281 .RS 15n
281 .RS 21n
282 This routine creates a service on top of an open and bound file descriptor, and
283 returns the handle to it. Typically, this descriptor is a connected file
284 descriptor for a connection-oriented transport. \fisendsz\fR and \fIrecvsz\fR
285 indicate sizes for the send and receive buffers. If they are \fB0\fR,
286 reasonable defaults are chosen. This routine returns \fINULL\fR if it fails,
287 and an error message is logged.
288 .RE

290 .sp
291 .ne 2
292 .na
293 \fB\fBsvc_raw_create()\fR\fR
294 .ad
295 .RS 15n
295 .RS 21n
295 This routine creates an \fBRPC\fR service handle and returns a pointer to it.
297 The transport is really a buffer within the process's address space, so the
298 corresponding \fBRPC\fR client should live in the same address space; (see
299 \fBcint_raw_create()\fR in \fBrpc_clnt_create\fR(3NSL)). This routine allows
300 simulation of \fBRPC\fR and acquisition of \fBRPC\fR overheads (such as round
301 trip times), without any kernel and networking interference. This routine
302 returns \fINULL\fR if it fails, and an error message is logged.
303 .sp
304 Note: \fBsvc_run()\fR should not be called when the raw interface is being
305 used.

```

```

306 .RE

308 .sp
309 .ne 2
310 .na
311 \fB\fBsvc_tli_create()\fR\fR
312 .ad
313 .RS 15n
313 .RS 21n
314 This routine creates an \fBRPC\fR server handle, and returns a pointer to it.
315 \fifildes\fR is the file descriptor on which the service is listening. If
316 \fifildes\fR is \fBRPC_ANYFD\fR, it opens a file descriptor on the transport
317 specified by \fInetconf\fR. If the file descriptor is unbound and
318 \fIbind_info\fR is non-null \fifildes\fR is bound to the address specified by
319 \fIbind_info\fR, otherwise \fifildes\fR is bound to a default address chosen by
320 \fIbindaddr\fR is non-null \fifildes\fR is bound to the address specified by
320 \fIbindaddr\fR, otherwise \fifildes\fR is bound to a default address chosen by
321 the transport. In the case where the default address is chosen, the number of
322 outstanding connect requests is set to 8 for connection-oriented transports.
323 The user may specify the size of the send and receive buffers with the
324 parameters \fIsendsz\fR and \fIrecvsz\fR \fIft\fR values of \fB0\fR choose
325 suitable defaults. This routine returns \fINULL\fR if it fails, and an error
326 message is logged. The server is not registered with the \fBrpcbind\fR(1M)
327 service.
327 .RE

329 .sp
330 .ne 2
331 .na
332 \fB\fBsvc_tp_create()\fR\fR
333 .ad
334 .RS 15n
324 .RS 21n
335 \fBsvc_tp_create()\fR creates a server handle for the network specified by
336 \fInetconf\fR, and registers itself with the \fBrpcbind\fR service.
337 \fIdispatch\fR is called when there is a remote procedure call for the given
338 \fIprogrnum\fR and \fIversnum\fR; this requires calling \fBsvc_run()\fR.
339 \fBsvc_tp_create()\fR returns the service handle if it succeeds, otherwise a
340 \fINULL\fR is returned and an error message is logged.
341 .RE

343 .sp
344 .ne 2
345 .na
346 \fB\fBsvc_tp_create_addr()\fR\fR
347 .ad
348 .RS 15n
349 \fBsvc_tp_create_addr()\fR creates a server handle for the network specified
350 by \fInetconf\fR, and registers itself with the \fBrpcbind\fR service.
351 If \fIbind_addr\fR is non-NULL, that address is used for the listener binding.
352 If \fIbind_addr\fR is NULL, this call is the same as \fBsvc_tp_create()\fR.
353 \fIdispatch\fR is called when there is a remote procedure call for the given
354 \fIprogrnum\fR and \fIversnum\fR; this requires calling \fBsvc_run()\fR.
355 \fBsvc_tp_create_addr()\fR returns the service handle if it succeeds,
356 otherwise a \fINULL\fR is returned and an error message is logged.
357 .RE

359 .sp
360 .ne 2
361 .na
362 \fB\fBsvc_vc_create()\fR\fR
363 .ad
364 .RS 15n
338 .RS 21n
365 This routine creates a connection-oriented \fBRPC\fR service and returns a
366 pointer to it. This routine returns \fINULL\fR if it fails, and an error

```

367 message is logged. The users may specify the size of the send and receive
368 buffers with the parameters \fIsendsz\fR and \fIrecvsz\fR; values of \fB0\fR
369 choose suitable defaults. The file descriptor \fIfildes\fR should be open and
370 bound. The server is not registered with the \fBrpcbind\fR(1M) service.
371 .RE

373 .sp
374 .ne 2
375 .na
376 \fB\fBsvc_door_create()\fR\fR
377 .ad
378 .RS 15n
352 .RS 21n
379 This routine creates an RPC server handle over doors for the given program
380 \fIprogram\fR and version \fIversnum\fR and returns a pointer to it.
381 Doors is a transport mechanism that facilitates fast data transfer between
354 \fIprogram\fR and version \fIversnum\fR and returns a pointer to
355 it. Doors is a transport mechanism that facilitates fast data transfer between
382 processes on the same machine. The user may set the size
383 of the send buffer with the parameter \fIsendsz\fR. If \fIsendsz\fR is 0, the
384 corresponding default buffer size is 16 Kbyte. If successful, the
385 \fBsvc_door_create()\fR routine returns the service handle. Otherwise it
386 returns \fINULL\fR and sets value for \fBrpc_createerr\fR. The server is not
387 registered with \fBrpcbind\fR(1M). The \fBSVCSET_CONNMAXREC\fR and
388 \fBSVCGET_CONNMAXREC\fR \fBsvc_control()\fR requests can be used to set and
389 change the maximum allowed request size for the doors transport.
390 .RE

392 .SH ATTRIBUTES
393 .LP
394 See \fBattributes\fR(5) for descriptions of the following attributes:
395 .sp

397 .sp
398 .TS
399 box;
400 c | c
401 l | l .
402 ATTRIBUTE TYPE ATTRIBUTE VALUE
403 _
404 Architecture All
405 _
406 Interface Stability Evolving
407 _
408 MT-Level MT-Safe
409 .TE

411 .SH SEE ALSO
412 .LP
413 \fBrpcbind\fR(1M), \fBrpc\fR(3NSL), \fBrpc_clnt_create\fR(3NSL),
414 \fBrpc_svc_calls\fR(3NSL), \fBrpc_svc_err\fR(3NSL), \fBrpc_svc_reg\fR(3NSL),
415 \fBattributes\fR(5)

new/usr/src/man/man4/nfs.4

7831 Mon Jul 17 16:15:19 2017

new/usr/src/man/man4/nfs.4

7569 statd support to run on a fixed port

Portions contributed by: Paul Dagnelie <pcd@delphix.com>

Reviewed by: Evan Layton <evan.layton@nexenta.com>

Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>

7577 mounted support to run on a fixed port

Portions contributed by: Paul Dagnelie <pcd@delphix.com>

Reviewed by: Evan Layton <evan.layton@nexenta.com>

Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>

1 .\"
2 .\" The contents of this file are subject to the terms of the
3 .\" Common Development and Distribution License (the "License").
4 .\" You may not use this file except in compliance with the License.
5 .\"
6 .\" You can obtain a copy of the license at `usr/src/OPENSOLARIS.LICENSE`
7 .\" or `http://www.opensolaris.org/os/licensing`.
8 .\" See the License for the specific language governing permissions
9 .\" and limitations under the License.
10 .\"
11 .\" When distributing Covered Code, include this CDDL HEADER in each
12 .\" file and include the License file at `usr/src/OPENSOLARIS.LICENSE`.
13 .\" If applicable, add the following below this CDDL HEADER, with the
14 .\" fields enclosed by brackets "[]" replaced with your own identifying
15 .\" information: Portions Copyright [yyyy] [name of copyright owner]
16 .\"
17 .\"
18 .\" Copyright 1989 AT&T
19 .\" Copyright (c) 2004, Sun Microsystems, Inc. All Rights Reserved.
20 .\" Copyright 2016 Nexenta Systems, Inc.
21 .\"
22 .Dd December 18, 2016
22 .Dd December 16, 2016
23 .Dt NFS 4
24 .Os
25 .Sh NAME
26 .Nm nfs
27 .Nd NFS configuration properties
28 .Sh DESCRIPTION
29 The behavior of the
30 .Xr nfssd 1M ,
31 .Xr nfsmapid 1M ,
32 .Xr lockd 1M ,
33 and
34 .Xr mountd 1M
35 daemons and
36 .Xr mount_nfs 1M
37 command is controlled by property values that are stored in the Service
38 Management Facility, `smf(5)`.
39 The
40 .Xr sharectl 1M
41 command should be used to query or change values for these properties.
42 .Pp
43 Changes made to
44 .Nm
45 property values on the
46 .Nm nfssd ,
47 .Nm lockd ,
48 .Nm mountd ,
49 or
50 .Nm mount_nfs
51 command line override the values set using
52 .Xr sharectl 1M .
53 .Pp

1

new/usr/src/man/man4/nfs.4

54 The following list describes the properties:

55 .Bl -tag -width Ds
56 .It Xo
57 .Sy client_versmin Ns = Ns Ar num
58 .br
59 .Sy client_versmax Ns = Ns Ar num
60 .Xc
61 The NFS client only uses NFS versions in the range specified by these
62 properties.
63 Valid values of versions are: 2, 3, and 4.
64 Default minimum version is
65 .Li 2 ,
66 while default maximum is
67 .Li 4 .
68 .Pp
69 You can override this range on a per-mount basis by using the
70 .Fl o Sy vers Ns =
71 option to
72 .Xr mount_nfs 1M .
73 .It Xo
74 .Sy server_versmin Ns = Ns Ar num
75 .br
76 .Sy server_versmax Ns = Ns Ar num
77 .Xc
78 The NFS server only uses NFS versions in the range specified by these
79 properties.
80 Valid values of versions are: 2, 3, and 4.
81 Default minimum version is
82 .Li 2 ,
83 while the default maximum version is
84 .Li 4 .
85 .It Sy server_delegation Ns = Ns Sy on Ns | Ns Sy off
86 By default the NFS server provides delegations to clients.
87 The user can turn off delegations for all exported filesystems by setting this
88 variable to
89 .Li off .
90 This variable only applies to NFS Version 4.
91 .It Sy nfsmapid_domain Ns = Ns Op Ar string
92 By default, the
93 .Nm nfsmapid
94 uses the DNS domain of the system.
95 This setting overrides the default.
96 This domain is used for identifying user and group attribute strings in the NFS
97 Version 4 protocol.
98 Clients and servers must match with this domain for operation to proceed
99 normally.
100 This variable only applies to NFS Version 4.
101 See
102 .Sx Setting nfsmapid_domain
103 below for further details.
104 .It Sy max_connections Ns = Ns Ar num
105 Sets the maximum number of concurrent, connection-oriented connections.
106 The default is
107 .Li -1
108 .Pq unlimited .
109 Equivalent to the
110 .Fl c
111 option in
112 .Nm nfssd .
113 .It Sy listen_backlog Ns = Ns Ar num
114 Set connection queue length for the NFS over a connection-oriented transport.
115 The default value is
116 .Li 32 '
117 meaning 32 entries in the queue.
118 Equivalent to the
119 .Fl l

2

```

120 option in
121 .Nm nfsd .
122 .It Sy protocol Ns = Ns Op Sy all Ns | Ns Ar protocol
123 Start
124 .Nm nfsd
125 over the specified protocol only.
126 Equivalent to the
127 .Fl p
128 option in
129 .Nm nfsd .
130 .Sy all
131 is equivalent to
132 .Fl a
133 on the
134 .Nm nfsd
135 command line.
136 Mutually exclusive of
137 .Sy device .
138 For the UDP protocol, only version 2 and version 3 service is established.
139 NFS Version 4 is not supported for the UDP protocol.
140 .It Sy device Ns = Ns Op Ar devname
141 Start NFS daemon for the transport specified by the given device only.
142 Equivalent to the
143 .Fl t
144 option in
145 .Nm nfsd .
146 Mutually exclusive of
147 .Sy protocol .
148 .It Sy servers Ns = Ns Ar num
149 Maximum number of concurrent NFS requests.
150 Equivalent to last numeric argument on the
151 .Nm nfsd
152 command line.
153 The default is
154 .Li 1024 .
155 .It Sy lockd_listen_backlog Ns = Ns Ar num
156 Set connection queue length for
157 .Nm lockd
158 over a connection-oriented transport.
159 The default and minimum value is
160 .Li 32 .
161 .It Sy lockd_servers Ns = Ns Ar num
162 Maximum number of concurrent
163 .Nm lockd
164 requests.
165 The default is 256.
166 .It Sy lockd_retransmit_timeout Ns = Ns Ar num
167 Retransmit timeout, in seconds, before
168 .Nm lockd
169 retries.
170 The default is
171 .Li 5 .
172 .It Sy grace_period Ns = Ns Ar num
173 Grace period, in seconds, that all clients
174 .Pq both NLM and NFSv4
175 have to reclaim locks after a server reboot.
176 This parameter also controls the NFSv4 lease interval.
177 The default is
178 .Li 90 .
179 .It Sy mountd_listen_backlog Ns = Ns Ar num
180 Set the connection queue length for
181 .Nm mountd
182 over a connection-oriented transport.
183 The default value is
184 .Li 64 .
185 .It Sy mountd_max_threads Ns = Ns Ar num

```

```

186 Maximum number of threads for
187 .Nm mountd .
188 The default value is
189 .Li 16 .
190 .It Sy mountd_port Ns = Ns Ar num
191 The IP port number on which
192 .Nm mountd
193 should listen.
194 The default value is
195 .Li 0 ,
196 which means it should use a default binding.
197 .It Sy statd_port Ns = Ns Ar num
198 The IP port number on which
199 .Nm statd
200 should listen.
201 The default value is
202 .Li 0 ,
203 which means it should use a default binding.
204 .El
205 .Ss Setting nfsmapid_domain
206 As described above, the setting for
207 .Sy nfsmapid_domain
208 overrides the domain used by
209 .Xr nfsmapid 1M
210 for building and comparing outbound and inbound attribute strings, respectively.
211 This setting overrides any other mechanism for setting the NFSv4 domain.
212 In the absence of a
213 .Sy nfsmapid_domain
214 setting, the
215 .Xr nfsmapid 1M
216 daemon determines the NFSv4 domain as follows:
217 .Bl -bullet
218 .It
219 If a properly configured
220 .Pa /etc/resolv.conf
221 .Po see
222 .Xr resolv.conf 4
223 .Pc
224 exists,
225 .Nm nfsmapid
226 queries specified nameserver(s) for the domain.
227 .It
228 If a properly configured
229 .Pa /etc/resolv.conf
230 .Po see
231 .Xr resolv.conf 4
232 .Pc
233 exists, but the queried nameserver does not have a proper record of the domain
234 name,
235 .Nm nfsmapid
236 attempts to obtain the domain name through the BIND interface
237 .Po see
238 .Xr resolver 3RESOLV
239 .Pc .
240 .It
241 If no
242 .Pa /etc/resolv.conf
243 exists,
244 .Nm nfsmapid
245 falls back on using the configured domain name
246 .Po see
247 .Xr domainname 1M
248 .Pc ,
249 which is returned with the leading domain suffix removed.
250 For example, for
251 .Li widgets.sales.acme.com ,

```

```
252 .Li sales.acme.com
253 is returned.
254 .It
255 If
256 .Pa /etc/resolv.conf
257 does not exist, no domain name has been configured
258 .Po or no
259 .Pa /etc/defaultdomain
260 exists
261 .Pc ,
262 .Nm nfsmapid
263 falls back on obtaining the domain name from the host name, if the host name
264 contains a fully qualified domain name
265 .Pg FQDN .
266 .El
267 .Pp
268 If a domainname is still not obtained following all of the preceding steps,
269 .Nm nfsmapid
270 will have no domain configured.
271 This results in the following behavior:
272 .Bl -bullet
273 .It
274 Outbound
275 .Qq owner
276 and
277 .Qq owner_group
278 attribute strings are encoded as literal id's.
279 For example, the UID 12345 is encoded as
280 .Li 12345 .
281 .It
282 .Nm nfsmapid
283 ignores the
284 .Qq domain
285 portion of the inbound attribute string and performs name service lookups only
286 for the user or group.
287 If the user/group exists in the local system name service databases, then the
288 proper uid/gid will be mapped even when no domain has been configured.
289 .Pp
290 This behavior implies that the same administrative user/group domain exists
291 between NFSv4 client and server (that is, the same uid/gid's for users/groups
292 on both client and server).
293 In the case of overlapping id spaces, the inbound attribute string could
294 potentially be mapped to the wrong id.
295 However, this is not functionally different from mapping the inbound string to
296 .Sy nobody'
297 yet provides greater flexibility.
298 .El
299 .Sh SEE ALSO
300 .Xr lockd 1M ,
301 .Xr mount_nfs 1M ,
302 .Xr mountd 1M ,
303 .Xr nfsd 1M ,
304 .Xr nfsmapid 1M ,
305 .Xr sharectl 1M ,
306 .Xr smf 5
```

new/usr/src/pkg/manifests/system-library.man3nsl.inc

20966 Mon Jul 17 16:15:20 2017

new/usr/src/pkg/manifests/system-library.man3nsl.inc

8330 Add svc_tp_create_addr to libnsl

Reviewed by: Paul Dagnelie <pcd@delphix.com>

Reviewed by: Evan Layton <evan.layton@nexenta.com>

Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License (" CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet
9 # at http://www.illumos.org/license/CDDL.
10 #

12 #
13 # Copyright 2011, Richard Lowe
14 # Copyright 2017 Nexenta Systems, Inc. All rights reserved.
14 # Copyright 2012 Nexenta Systems, Inc. All rights reserved.
15 #

17 file path=usr/share/man/man3nsl/dial.3nsl
18 file path=usr/share/man/man3nsl/doconfig.3nsl
19 file path=usr/share/man/man3nsl/gethostbyname.3nsl
20 file path=usr/share/man/man3nsl/getipsecalgbyname.3nsl
21 file path=usr/share/man/man3nsl/getipsecprotobyname.3nsl
22 file path=usr/share/man/man3nsl/getnetconfig.3nsl
23 file path=usr/share/man/man3nsl/getnetpath.3nsl
24 file path=usr/share/man/man3nsl/getpublickey.3nsl
25 file path=usr/share/man/man3nsl/getrpcbyname.3nsl
26 file path=usr/share/man/man3nsl/netdir.3nsl
27 file path=usr/share/man/man3nsl/nlsgetcall.3nsl
28 file path=usr/share/man/man3nsl/nlsprovider.3nsl
29 file path=usr/share/man/man3nsl/nlsrequest.3nsl
30 file path=usr/share/man/man3nsl/rpc.3nsl
31 file path=usr/share/man/man3nsl/rpc_clnt_auth.3nsl
32 file path=usr/share/man/man3nsl/rpc_clnt_calls.3nsl
33 file path=usr/share/man/man3nsl/rpc_clnt_create.3nsl
34 file path=usr/share/man/man3nsl/rpc_control.3nsl
35 file path=usr/share/man/man3nsl/rpc_gss_get_error.3nsl
36 file path=usr/share/man/man3nsl/rpc_gss_get_mechanisms.3nsl
37 file path=usr/share/man/man3nsl/rpc_gss_get_principal_name.3nsl
38 file path=usr/share/man/man3nsl/rpc_gss_getcred.3nsl
39 file path=usr/share/man/man3nsl/rpc_gss_max_data_length.3nsl
40 file path=usr/share/man/man3nsl/rpc_gss_mech_to_oid.3nsl
41 file path=usr/share/man/man3nsl/rpc_gss_seccreate.3nsl
42 file path=usr/share/man/man3nsl/rpc_gss_set_callback.3nsl
43 file path=usr/share/man/man3nsl/rpc_gss_set_defaults.3nsl
44 file path=usr/share/man/man3nsl/rpc_gss_set_svc_name.3nsl
45 file path=usr/share/man/man3nsl/rpc_soc.3nsl
46 file path=usr/share/man/man3nsl/rpc_svc_calls.3nsl
47 file path=usr/share/man/man3nsl/rpc_svc_create.3nsl
48 file path=usr/share/man/man3nsl/rpc_svc_err.3nsl
49 file path=usr/share/man/man3nsl/rpc_svc_input.3nsl
50 file path=usr/share/man/man3nsl/rpc_svc_reg.3nsl
51 file path=usr/share/man/man3nsl/rpc_xdr.3nsl
52 file path=usr/share/man/man3nsl/rpcbind.3nsl
53 file path=usr/share/man/man3nsl/rpcsec_gss.3nsl
54 file path=usr/share/man/man3nsl/secure_rpc.3nsl
55 file path=usr/share/man/man3nsl/t_accept.3nsl
56 file path=usr/share/man/man3nsl/t_alloc.3nsl
57 file path=usr/share/man/man3nsl/t_bind.3nsl
```

1

new/usr/src/pkg/manifests/system-library.man3nsl.inc

```
58 file path=usr/share/man/man3nsl/t_close.3nsl
59 file path=usr/share/man/man3nsl/t_connect.3nsl
60 file path=usr/share/man/man3nsl/t_errno.3nsl
61 file path=usr/share/man/man3nsl/t_error.3nsl
62 file path=usr/share/man/man3nsl/t_free.3nsl
63 file path=usr/share/man/man3nsl/t_getinfo.3nsl
64 file path=usr/share/man/man3nsl/t_getprotaddr.3nsl
65 file path=usr/share/man/man3nsl/t_getstate.3nsl
66 file path=usr/share/man/man3nsl/t_listen.3nsl
67 file path=usr/share/man/man3nsl/t_look.3nsl
68 file path=usr/share/man/man3nsl/t_open.3nsl
69 file path=usr/share/man/man3nsl/t_optmgmt.3nsl
70 file path=usr/share/man/man3nsl/t_rcv.3nsl
71 file path=usr/share/man/man3nsl/t_rcvconnect.3nsl
72 file path=usr/share/man/man3nsl/t_rcvdis.3nsl
73 file path=usr/share/man/man3nsl/t_rcvrel.3nsl
74 file path=usr/share/man/man3nsl/t_rcvreldata.3nsl
75 file path=usr/share/man/man3nsl/t_rcvdata.3nsl
76 file path=usr/share/man/man3nsl/t_rcvuderr.3nsl
77 file path=usr/share/man/man3nsl/t_rcv.3nsl
78 file path=usr/share/man/man3nsl/t_rcvvudata.3nsl
79 file path=usr/share/man/man3nsl/t_snd.3nsl
80 file path=usr/share/man/man3nsl/t_snddis.3nsl
81 file path=usr/share/man/man3nsl/t_sndrel.3nsl
82 file path=usr/share/man/man3nsl/t_sndreldata.3nsl
83 file path=usr/share/man/man3nsl/t_sndudata.3nsl
84 file path=usr/share/man/man3nsl/t_sndv.3nsl
85 file path=usr/share/man/man3nsl/t_sndvudata.3nsl
86 file path=usr/share/man/man3nsl/t_strerror.3nsl
87 file path=usr/share/man/man3nsl/t_sync.3nsl
88 file path=usr/share/man/man3nsl/t_sysconf.3nsl
89 file path=usr/share/man/man3nsl/t_unbind.3nsl
90 file path=usr/share/man/man3nsl/xdr.3nsl
91 file path=usr/share/man/man3nsl/xdr_admin.3nsl
92 file path=usr/share/man/man3nsl/xdr_complex.3nsl
93 file path=usr/share/man/man3nsl/xdr_create.3nsl
94 file path=usr/share/man/man3nsl/xdr_simple.3nsl
95 file path=usr/share/man/man3nsl/yp_update.3nsl
96 file path=usr/share/man/man3nsl/ypclnt.3nsl
97 link path=usr/share/man/man3nsl/auth_destroy.3nsl target=rpc_clnt_auth.3nsl
98 link path=usr/share/man/man3nsl/authdes_create.3nsl target=rpc_soc.3nsl
99 link path=usr/share/man/man3nsl/authdes_getcrec.3nsl target=secure_rpc.3nsl
100 link path=usr/share/man/man3nsl/authdes_seccreate.3nsl target=secure_rpc.3nsl
101 link path=usr/share/man/man3nsl/authnone_create.3nsl target=rpc_clnt_auth.3nsl
102 link path=usr/share/man/man3nsl/authsys_create.3nsl target=rpc_clnt_auth.3nsl
103 link path=usr/share/man/man3nsl/authsys_create_default.3nsl \
104     target=rpc_clnt_auth.3nsl
105 link path=usr/share/man/man3nsl/authunix_create.3nsl target=rpc_soc.3nsl
106 link path=usr/share/man/man3nsl/authunix_create_default.3nsl \
107     target=rpc_soc.3nsl
108 link path=usr/share/man/man3nsl/callrpc.3nsl target=rpc_soc.3nsl
109 link path=usr/share/man/man3nsl/clnt_broadcast.3nsl target=rpc_soc.3nsl
110 link path=usr/share/man/man3nsl/clnt_call.3nsl target=rpc_clnt_calls.3nsl
111 link path=usr/share/man/man3nsl/clnt_control.3nsl target=rpc_clnt_create.3nsl
112 link path=usr/share/man/man3nsl/clnt_create.3nsl target=rpc_clnt_create.3nsl
113 link path=usr/share/man/man3nsl/clnt_create_timed.3nsl \
114     target=rpc_clnt_create.3nsl
115 link path=usr/share/man/man3nsl/clnt_create_vers.3nsl \
116     target=rpc_clnt_create.3nsl
117 link path=usr/share/man/man3nsl/clnt_create_vers_timed.3nsl \
118     target=rpc_clnt_create.3nsl
119 link path=usr/share/man/man3nsl/clnt_destroy.3nsl target=rpc_clnt_create.3nsl
120 link path=usr/share/man/man3nsl/clnt_dg_create.3nsl \
121     target=rpc_clnt_create.3nsl
122 link path=usr/share/man/man3nsl/clnt_door_create.3nsl \
123     target=rpc_clnt_create.3nsl
```

2

```

124 link path=usr/share/man/man3nsl/clnt_freeres.3nsl target=rpc_clnt_calls.3nsl
125 link path=usr/share/man/man3nsl/clnt_geterr.3nsl target=rpc_clnt_calls.3nsl
126 link path=usr/share/man/man3nsl/clnt_pcreateerror.3nsl \
127     target=rpc_clnt_create.3nsl
128 link path=usr/share/man/man3nsl/clnt_perrno.3nsl target=rpc_clnt_calls.3nsl
129 link path=usr/share/man/man3nsl/clnt_perror.3nsl target=rpc_clnt_calls.3nsl
130 link path=usr/share/man/man3nsl/clnt_raw_create.3nsl \
131     target=rpc_clnt_create.3nsl
132 link path=usr/share/man/man3nsl/clnt_send.3nsl target=rpc_clnt_calls.3nsl
133 link path=usr/share/man/man3nsl/clnt_spcreateerror.3nsl \
134     target=rpc_clnt_create.3nsl
135 link path=usr/share/man/man3nsl/clnt_serrno.3nsl target=rpc_clnt_calls.3nsl
136 link path=usr/share/man/man3nsl/clnt_serror.3nsl target=rpc_clnt_calls.3nsl
137 link path=usr/share/man/man3nsl/clnt_tli_create.3nsl \
138     target=rpc_clnt_create.3nsl
139 link path=usr/share/man/man3nsl/clnt_tp_create.3nsl \
140     target=rpc_clnt_create.3nsl
141 link path=usr/share/man/man3nsl/clnt_tp_create_timed.3nsl \
142     target=rpc_clnt_create.3nsl
143 link path=usr/share/man/man3nsl/clnt_vc_create.3nsl \
144     target=rpc_clnt_create.3nsl
145 link path=usr/share/man/man3nsl/clntraw_create.3nsl target=rpc_soc.3nsl
146 link path=usr/share/man/man3nsl/clnttcp_create.3nsl target=rpc_soc.3nsl
147 link path=usr/share/man/man3nsl/clntudp_bufcreate.3nsl target=rpc_soc.3nsl
148 link path=usr/share/man/man3nsl/clntudp_create.3nsl target=rpc_soc.3nsl
149 link path=usr/share/man/man3nsl/endhostbyname.3nsl target=gethostbyname.3nsl
150 link path=usr/share/man/man3nsl/endnetconfig.3nsl target=getnetconfig.3nsl
151 link path=usr/share/man/man3nsl/endnetpath.3nsl target=getnetpath.3nsl
152 link path=usr/share/man/man3nsl/endrpcnt.3nsl target=getrpbyname.3nsl
153 link path=usr/share/man/man3nsl/freeipsecalgent.3nsl \
154     target=getipsecalgbyname.3nsl
155 link path=usr/share/man/man3nsl/freenetconfigent.3nsl target=getnetconfig.3nsl
156 link path=usr/share/man/man3nsl/get_myaddress.3nsl target=rpc_soc.3nsl
157 link path=usr/share/man/man3nsl/gethostbyaddr.3nsl target=gethostbyname.3nsl
158 link path=usr/share/man/man3nsl/gethostbyaddr_r.3nsl target=gethostbyname.3nsl
159 link path=usr/share/man/man3nsl/gethostbyname_r.3nsl target=gethostbyname.3nsl
160 link path=usr/share/man/man3nsl/gethostent.3nsl target=gethostbyname.3nsl
161 link path=usr/share/man/man3nsl/gethostent_r.3nsl target=gethostbyname.3nsl
162 link path=usr/share/man/man3nsl/getipsecalgbynum.3nsl \
163     target=getipsecalgbyname.3nsl
164 link path=usr/share/man/man3nsl/getipsecproto bynum.3nsl \
165     target=getipsecproto byname.3nsl
166 link path=usr/share/man/man3nsl/getnetconfigent.3nsl target=getnetconfig.3nsl
167 link path=usr/share/man/man3nsl/getnetname.3nsl target=secure_rpc.3nsl
168 link path=usr/share/man/man3nsl/getrpbyname_r.3nsl target=getrpbyname.3nsl
169 link path=usr/share/man/man3nsl/getrpbynumber.3nsl target=getrpbyname.3nsl
170 link path=usr/share/man/man3nsl/getrpbynunber_r.3nsl target=getrpbyname.3nsl
171 link path=usr/share/man/man3nsl/getrpcnt.3nsl target=getrpbyname.3nsl
172 link path=usr/share/man/man3nsl/getrpcnt_r.3nsl target=getrpbyname.3nsl
173 link path=usr/share/man/man3nsl/getrpcport.3nsl target=rpc_soc.3nsl
174 link path=usr/share/man/man3nsl/getsecretkey.3nsl target=getpublickey.3nsl
175 link path=usr/share/man/man3nsl/host2netname.3nsl target=secure_rpc.3nsl
176 link path=usr/share/man/man3nsl/key_decryptsession.3nsl target=secure_rpc.3nsl
177 link path=usr/share/man/man3nsl/key_encryptsession.3nsl target=secure_rpc.3nsl
178 link path=usr/share/man/man3nsl/key_gendes.3nsl target=secure_rpc.3nsl
179 link path=usr/share/man/man3nsl/key_secrectkey_is_set.3nsl \
180     target=secure_rpc.3nsl
181 link path=usr/share/man/man3nsl/key_setsecret.3nsl target=secure_rpc.3nsl
182 link path=usr/share/man/man3nsl/nc_perror.3nsl target=getnetconfig.3nsl
183 link path=usr/share/man/man3nsl/nc_serror.3nsl target=getnetconfig.3nsl
184 link path=usr/share/man/man3nsl/netdir_free.3nsl target=netdir.3nsl
185 link path=usr/share/man/man3nsl/netdir_getbyaddr.3nsl target=netdir.3nsl
186 link path=usr/share/man/man3nsl/netdir_getbyname.3nsl target=netdir.3nsl
187 link path=usr/share/man/man3nsl/netdir_mergeaddr.3nsl target=netdir.3nsl
188 link path=usr/share/man/man3nsl/netdir_options.3nsl target=netdir.3nsl
189 link path=usr/share/man/man3nsl/netdir_perror.3nsl target=netdir.3nsl

```

```

190 link path=usr/share/man/man3nsl/netdir_sperror.3nsl target=netdir.3nsl
191 link path=usr/share/man/man3nsl/netname2host.3nsl target=secure_rpc.3nsl
192 link path=usr/share/man/man3nsl/netname2user.3nsl target=secure_rpc.3nsl
193 link path=usr/share/man/man3nsl/pmap_getmaps.3nsl target=rpc_soc.3nsl
194 link path=usr/share/man/man3nsl/pmap_getport.3nsl target=rpc_soc.3nsl
195 link path=usr/share/man/man3nsl/pmap_rmtcall.3nsl target=rpc_soc.3nsl
196 link path=usr/share/man/man3nsl/pmap_set.3nsl target=rpc_soc.3nsl
197 link path=usr/share/man/man3nsl/pmap_unset.3nsl target=rpc_soc.3nsl
198 link path=usr/share/man/man3nsl/publickey.3nsl target=getpublickey.3nsl
199 link path=usr/share/man/man3nsl/registerrpc.3nsl target=rpc_soc.3nsl
200 link path=usr/share/man/man3nsl/rpc_broadcast.3nsl target=rpc_clnt_calls.3nsl
201 link path=usr/share/man/man3nsl/rpc_broadcast_exp.3nsl \
202     target=rpc_clnt_calls.3nsl
203 link path=usr/share/man/man3nsl/rpc_call.3nsl target=rpc_clnt_calls.3nsl
204 link path=usr/share/man/man3nsl/rpc_createerr.3nsl target=rpc_clnt_create.3nsl
205 link path=usr/share/man/man3nsl/rpc_gss_get_mech_info.3nsl \
206     target=rpc_gss_get_mechanisms.3nsl
207 link path=usr/share/man/man3nsl/rpc_gss_get_versions.3nsl \
208     target=rpc_gss_get_mechanisms.3nsl
209 link path=usr/share/man/man3nsl/rpc_gss_is_installed.3nsl \
210     target=rpc_gss_get_mechanisms.3nsl
211 link path=usr/share/man/man3nsl/rpc_gss_qop_to_num.3nsl \
212     target=rpc_gss_mech_to_oid.3nsl
213 link path=usr/share/man/man3nsl/rpc_gss_svc_max_data_length.3nsl \
214     target=rpc_gss_max_data_length.3nsl
215 link path=usr/share/man/man3nsl/rpc_reg.3nsl target=rpc_svc_reg.3nsl
216 link path=usr/share/man/man3nsl/rpcb_getaddr.3nsl target=rpcbind.3nsl
217 link path=usr/share/man/man3nsl/rpcb_getmaps.3nsl target=rpcbind.3nsl
218 link path=usr/share/man/man3nsl/rpcb_gettime.3nsl target=rpcbind.3nsl
219 link path=usr/share/man/man3nsl/rpcb_rmtcall.3nsl target=rpcbind.3nsl
220 link path=usr/share/man/man3nsl/rpcb_set.3nsl target=rpcbind.3nsl
221 link path=usr/share/man/man3nsl/rpcb_unset.3nsl target=rpcbind.3nsl
222 link path=usr/share/man/man3nsl/sethostent.3nsl target=gethostbyname.3nsl
223 link path=usr/share/man/man3nsl/setnetconfig.3nsl target=getnetconfig.3nsl
224 link path=usr/share/man/man3nsl/setnetpath.3nsl target=getnetpath.3nsl
225 link path=usr/share/man/man3nsl/setrpcent.3nsl target=getrpbyname.3nsl
226 link path=usr/share/man/man3nsl/svc_add_input.3nsl target=rpc_svc_input.3nsl
227 link path=usr/share/man/man3nsl/svc_auth_reg.3nsl target=rpc_svc_reg.3nsl
228 link path=usr/share/man/man3nsl/svc_control.3nsl target=rpc_svc_create.3nsl
229 link path=usr/share/man/man3nsl/svc_create.3nsl target=rpc_svc_create.3nsl
230 link path=usr/share/man/man3nsl/svc_destroy.3nsl target=rpc_svc_create.3nsl
231 link path=usr/share/man/man3nsl/svc_dg_create.3nsl target=rpc_svc_create.3nsl
232 link path=usr/share/man/man3nsl/svc_dg_enablecache.3nsl \
233     target=rpc_svc_calls.3nsl
234 link path=usr/share/man/man3nsl/svc_done.3nsl target=rpc_svc_calls.3nsl
235 link path=usr/share/man/man3nsl/svc_door_create.3nsl \
236     target=rpc_svc_create.3nsl
237 link path=usr/share/man/man3nsl/svc_exit.3nsl target=rpc_svc_calls.3nsl
238 link path=usr/share/man/man3nsl/svc_fd_create.3nsl target=rpc_svc_create.3nsl
239 link path=usr/share/man/man3nsl/svc_fd_negotiate_ucred.3nsl \
240     target=rpc_svc_calls.3nsl
241 link path=usr/share/man/man3nsl/svc_fds.3nsl target=rpc_soc.3nsl
242 link path=usr/share/man/man3nsl/svc_fdset.3nsl target=rpc_svc_calls.3nsl
243 link path=usr/share/man/man3nsl/svc_freeargs.3nsl target=rpc_svc_calls.3nsl
244 link path=usr/share/man/man3nsl/svc_getargs.3nsl target=rpc_svc_calls.3nsl
245 link path=usr/share/man/man3nsl/svc_getcaller.3nsl target=rpc_soc.3nsl
246 link path=usr/share/man/man3nsl/svc_getcallerucred.3nsl \
247     target=rpc_svc_calls.3nsl
248 link path=usr/share/man/man3nsl/svc_getreq.3nsl target=rpc_soc.3nsl
249 link path=usr/share/man/man3nsl/svc_getreq_common.3nsl \
250     target=rpc_svc_calls.3nsl
251 link path=usr/share/man/man3nsl/svc_getreq_poll.3nsl target=rpc_svc_calls.3nsl
252 link path=usr/share/man/man3nsl/svc_getreqset.3nsl target=rpc_svc_calls.3nsl
253 link path=usr/share/man/man3nsl/svc_getrpccaller.3nsl \
254     target=rpc_svc_calls.3nsl
255 link path=usr/share/man/man3nsl/svc_max_pollfd.3nsl target=rpc_svc_calls.3nsl

```

```

256 link path=usr/share/man/man3nsl/svc_pollfd.3nsl target=rpc_svc_calls.3nsl
257 link path=usr/share/man/man3nsl/svc_raw_create.3nsl target=rpc_svc_create.3nsl
258 link path=usr/share/man/man3nsl/svc_reg.3nsl target=rpc_svc_reg.3nsl
259 link path=usr/share/man/man3nsl/svc_register.3nsl target=rpc_soc.3nsl
260 link path=usr/share/man/man3nsl/svc_remove_input.3nsl \
261     target=rpc_svc_input.3nsl
262 link path=usr/share/man/man3nsl/svc_run.3nsl target=rpc_svc_calls.3nsl
263 link path=usr/share/man/man3nsl/svc_sendreply.3nsl target=rpc_svc_calls.3nsl
264 link path=usr/share/man/man3nsl/svc_tli_create.3nsl target=rpc_svc_create.3nsl
265 link path=usr/share/man/man3nsl/svc_tp_create.3nsl target=rpc_svc_create.3nsl
266 link path=usr/share/man/man3nsl/svc_tp_create_addr.3nsl \
267     target=rpc_svc_create.3nsl
268 link path=usr/share/man/man3nsl/svc_unreg.3nsl target=rpc_svc_reg.3nsl
269 link path=usr/share/man/man3nsl/svc_unregister.3nsl target=rpc_soc.3nsl
270 link path=usr/share/man/man3nsl/svc_vc_create.3nsl target=rpc_svc_create.3nsl
271 link path=usr/share/man/man3nsl/svcerr_auth.3nsl target=rpc_svc_err.3nsl
272 link path=usr/share/man/man3nsl/svcerr_decode.3nsl target=rpc_svc_err.3nsl
273 link path=usr/share/man/man3nsl/svcerr_noproc.3nsl target=rpc_svc_err.3nsl
274 link path=usr/share/man/man3nsl/svcerr_noprog.3nsl target=rpc_svc_err.3nsl
275 link path=usr/share/man/man3nsl/svcerr_proovers.3nsl target=rpc_svc_err.3nsl
276 link path=usr/share/man/man3nsl/svcerr_systemerr.3nsl target=rpc_svc_err.3nsl
277 link path=usr/share/man/man3nsl/svcerr_weakauth.3nsl target=rpc_svc_err.3nsl
278 link path=usr/share/man/man3nsl/svcfd_create.3nsl target=rpc_soc.3nsl
279 link path=usr/share/man/man3nsl/svcraw_create.3nsl target=rpc_soc.3nsl
280 link path=usr/share/man/man3nsl/svctcp_create.3nsl target=rpc_soc.3nsl
281 link path=usr/share/man/man3nsl/svcudp_bufcreate.3nsl target=rpc_soc.3nsl
282 link path=usr/share/man/man3nsl/svcudp_create.3nsl target=rpc_soc.3nsl
283 link path=usr/share/man/man3nsl/taddr2uaddr.3nsl target=netdir.3nsl
284 link path=usr/share/man/man3nsl/uaddr2taddr.3nsl target=netdir.3nsl
285 link path=usr/share/man/man3nsl/undai.3nsl target=dial.3nsl
286 link path=usr/share/man/man3nsl/user2netname.3nsl target=secure_rpc.3nsl
287 link path=usr/share/man/man3nsl/xdr_accepted_reply.3nsl target=rpc_xdr.3nsl
288 link path=usr/share/man/man3nsl/xdr_array.3nsl target=xdr_complex.3nsl
289 link path=usr/share/man/man3nsl/xdr_authsys_parms.3nsl target=rpc_xdr.3nsl
290 link path=usr/share/man/man3nsl/xdr_authunix_parms.3nsl target=rpc_soc.3nsl
291 link path=usr/share/man/man3nsl/xdr_bool.3nsl target=xdr_simple.3nsl
292 link path=usr/share/man/man3nsl/xdr_bytes.3nsl target=xdr_complex.3nsl
293 link path=usr/share/man/man3nsl/xdr_callhdr.3nsl target=rpc_xdr.3nsl
294 link path=usr/share/man/man3nsl/xdr_callmsg.3nsl target=rpc_xdr.3nsl
295 link path=usr/share/man/man3nsl/xdr_char.3nsl target=xdr_simple.3nsl
296 link path=usr/share/man/man3nsl/xdr_control.3nsl target=xdr_admin.3nsl
297 link path=usr/share/man/man3nsl/xdr_destroy.3nsl target=xdr_create.3nsl
298 link path=usr/share/man/man3nsl/xdr_double.3nsl target=xdr_simple.3nsl
299 link path=usr/share/man/man3nsl/xdr_enum.3nsl target=xdr_simple.3nsl
300 link path=usr/share/man/man3nsl/xdr_float.3nsl target=xdr_simple.3nsl
301 link path=usr/share/man/man3nsl/xdr_free.3nsl target=xdr_simple.3nsl
302 link path=usr/share/man/man3nsl/xdr_getpos.3nsl target=xdr_admin.3nsl
303 link path=usr/share/man/man3nsl/xdr_hyper.3nsl target=xdr_simple.3nsl
304 link path=usr/share/man/man3nsl/xdr_inline.3nsl target=xdr_admin.3nsl
305 link path=usr/share/man/man3nsl/xdr_int.3nsl target=xdr_simple.3nsl
306 link path=usr/share/man/man3nsl/xdr_long.3nsl target=xdr_simple.3nsl
307 link path=usr/share/man/man3nsl/xdr_longlong_t.3nsl target=xdr_simple.3nsl
308 link path=usr/share/man/man3nsl/xdr_opaque.3nsl target=xdr_complex.3nsl
309 link path=usr/share/man/man3nsl/xdr_opaque_auth.3nsl target=rpc_xdr.3nsl
310 link path=usr/share/man/man3nsl/xdr_pointer.3nsl target=xdr_complex.3nsl
311 link path=usr/share/man/man3nsl/xdr_quadruple.3nsl target=xdr_simple.3nsl
312 link path=usr/share/man/man3nsl/xdr_reference.3nsl target=xdr_complex.3nsl
313 link path=usr/share/man/man3nsl/xdr_rejected_reply.3nsl target=rpc_xdr.3nsl
314 link path=usr/share/man/man3nsl/xdr_repliesmsg.3nsl target=rpc_xdr.3nsl
315 link path=usr/share/man/man3nsl/xdr_setpos.3nsl target=xdr_admin.3nsl
316 link path=usr/share/man/man3nsl/xdr_short.3nsl target=xdr_simple.3nsl
317 link path=usr/share/man/man3nsl/xdr_sizeof.3nsl target=xdr_admin.3nsl
318 link path=usr/share/man/man3nsl/xdr_string.3nsl target=xdr_complex.3nsl
319 link path=usr/share/man/man3nsl/xdr_u_char.3nsl target=xdr_simple.3nsl
320 link path=usr/share/man/man3nsl/xdr_u_hyper.3nsl target=xdr_simple.3nsl
321 link path=usr/share/man/man3nsl/xdr_u_int.3nsl target=xdr_simple.3nsl

```

```

322 link path=usr/share/man/man3nsl/xdr_u_long.3nsl target=xdr_simple.3nsl
323 link path=usr/share/man/man3nsl/xdr_u_longlong_t.3nsl target=xdr_simple.3nsl
324 link path=usr/share/man/man3nsl/xdr_u_short.3nsl target=xdr_simple.3nsl
325 link path=usr/share/man/man3nsl/xdr_union.3nsl target=xdr_complex.3nsl
326 link path=usr/share/man/man3nsl/xdr_vector.3nsl target=xdr_complex.3nsl
327 link path=usr/share/man/man3nsl/xdr_void.3nsl target=xdr_simple.3nsl
328 link path=usr/share/man/man3nsl/xdr_wrapstring.3nsl target=xdr_complex.3nsl
329 link path=usr/share/man/man3nsl/xdrmem_create.3nsl target=xdr_create.3nsl
330 link path=usr/share/man/man3nsl/xdrrec_create.3nsl target=xdr_create.3nsl
331 link path=usr/share/man/man3nsl/xdrrec_endofrecord.3nsl target=xdr_admin.3nsl
332 link path=usr/share/man/man3nsl/xdrrec_eof.3nsl target=xdr_admin.3nsl
333 link path=usr/share/man/man3nsl/xdrrec_readbytes.3nsl target=xdr_admin.3nsl
334 link path=usr/share/man/man3nsl/xdrrec_skiprecord.3nsl target=xdr_admin.3nsl
335 link path=usr/share/man/man3nsl/xdrstdio_create.3nsl target=xdr_create.3nsl
336 link path=usr/share/man/man3nsl/xprt_register.3nsl target=rpc_svc_reg.3nsl
337 link path=usr/share/man/man3nsl/xprt_unregister.3nsl target=rpc_svc_reg.3nsl
338 link path=usr/share/man/man3nsl/yp_all.3nsl target=ypclnt.3nsl
339 link path=usr/share/man/man3nsl/yp_bind.3nsl target=ypclnt.3nsl
340 link path=usr/share/man/man3nsl/yp_first.3nsl target=ypclnt.3nsl
341 link path=usr/share/man/man3nsl/yp_get_default_domain.3nsl target=ypclnt.3nsl
342 link path=usr/share/man/man3nsl/yp_master.3nsl target=ypclnt.3nsl
343 link path=usr/share/man/man3nsl/yp_match.3nsl target=ypclnt.3nsl
344 link path=usr/share/man/man3nsl/yp_next.3nsl target=ypclnt.3nsl
345 link path=usr/share/man/man3nsl/yp_order.3nsl target=ypclnt.3nsl
346 link path=usr/share/man/man3nsl/yp_unbind.3nsl target=ypclnt.3nsl
347 link path=usr/share/man/man3nsl/yperr_string.3nsl target=ypclnt.3nsl
348 link path=usr/share/man/man3nsl/ypprot_err.3nsl target=ypclnt.3nsl

```

new/usr/src/uts/common/rpc/svc.h

37388 Mon Jul 17 16:15:20 2017

new/usr/src/uts/common/rpc/svc.h

8330 Add svc_tp_create_addr to libns1

Reviewed by: Paul Dagnelie <pcd@delphix.com>

Reviewed by: Evan Layton <evan.layton@nexenta.com>

Reviewed by: Sebastien Roy <sebastien.roy@delphix.com>

_____ unchanged_portion_omitted _____

```
826 extern int      svc_rdma_kcreate(char *, SVC_CALLOUT_TABLE *, int,
827                  rdma_xprt_group_t *);
828 extern void     svc_rdma_kstop(SVCMASTERXPRT *);
829 extern void     svc_rdma_kdestroy(SVCMASTERXPRT *);
830 extern void     rdma_stop(rdma_xprt_group_t *);

832 /*
833  * GSS cleanup method.
834 */
835 extern void     rpc_gss_cleanup(SVCXPRT *);
836 #else /* _KERNEL */
837 /*
838  * Lowest level dispatching -OR- who owns this process anyway.
839  * Somebody has to wait for incoming requests and then call the correct
840  * service routine. The routine svc_run does infinite waiting; i.e.,
841  * svc_run never returns.
842  * Since another (co-existant) package may wish to selectively wait for
843  * incoming calls or other events outside of the rpc architecture, the
844  * routine svc_getreq_poll is provided. It must be passed pollfds, the
845  * "in-place" results of a poll call (see poll, section 2).
846 */

848 /*
849  * Global keeper of rpc service descriptors in use
850  * dynamic; must be inspected before each call to select or poll
851 */
852 extern pollfd_t *svc_pollfd;
853 extern int      svc_max_pollfd;
854 extern fd_set   svc_fdset;
855 #define svc_fds  svc_fdset.fds_bits[0] /* compatibility */

857 /*
858  * A small program implemented by the svc_rpc implementation itself.
859  * Also see clnt.h for protocol numbers.
860 */
861 #ifdef __STDC__
862 extern void     svc_getreq(int);
863 extern void     svc_getreq_common(const int);
864 extern void     svc_getreqset(fd_set *); /* takes fdset instead of int */
865 extern void     svc_getreq_poll(struct pollfd *, const int);
866 extern void     svc_run(void);
867 extern void     svc_exit(void);
868 #else /* __STDC__ */
869 extern void     rpctest_service();
870 extern void     svc_getreqset();
871 extern void     svc_getreq();
872 extern void     svc_getreq_common();
873 extern void     svc_getreqset(); /* takes fdset instead of int */
874 extern void     svc_getreq_poll();
875 extern void     svc_run();
876 extern void     svc_exit();
877#endif /* __STDC__ */

879 /*
880  * Functions used to manage user file descriptors
881 */

```

1

new/usr/src/uts/common/rpc/svc.h

```
882 typedef int svc_input_id_t;
883 typedef void (*svc_callback_t)(svc_input_id_t id, int fd,
884                               unsigned int events, void* cookie);

886 #ifdef __STDC__
887 extern svc_input_id_t svc_add_input(int fd, unsigned int events,
888                                     svc_callback_t user_callback,
889                                     void* cookie);
890 extern int svc_remove_input(svc_input_id_t id);
891 #else /* __STDC__ */
892 extern svc_input_id_t svc_add_input();
893 extern int svc_remove_input();
894#endif

896 /*
897  * These are the existing service side transport implementations.
898 */
899 /* Transport independent svc_create routine.
900 */
901 #ifdef __STDC__
902 extern int svc_create(void (*)(struct svc_req *, SVCXPRT *),
903                      const rpcprog_t, const rpcvers_t,
904                      const char *);
905 /*
906  * void (*dispatch)();
907  * const rpcprog_t progrnum; -- dispatch routine
908  * const rpcvers_t versnum; -- program number
909  * const char *nettype; -- version number
910  */
912 /*
913  * Generic server creation routine. It takes a netconfig structure
914  * instead of a nettype.
915 */
916 extern SVCXPRT *svc_tp_create(void (*)(struct svc_req *, SVCXPRT *),
917                                const rpcprog_t, const rpcvers_t,
918                                const struct netconfig *);
919 /*
920  * void (*dispatch)();
921  * const rpcprog_t progrnum; -- dispatch routine
922  * const rpcvers_t versnum; -- program number
923  * const struct netconfig *nconf; -- version number
924  */
926 /*
927  * Variant of svc_tp_create that accepts a binding address.
928  * If addr == NULL, this is the same as svc_tp_create().
929 */
930 extern SVCXPRT *svc_tp_create_addr(void (*)(struct svc_req *, SVCXPRT *),
931                                     const rpcprog_t, const rpcvers_t,
932                                     const struct netconfig *,
933                                     const struct netbuf *);
934 /*
935  * void (*dispatch)();
936  * const rpcprog_t progrnum; -- dispatch routine
937  * const rpcvers_t versnum; -- program number
938  * const struct netconfig *nconf; -- version number
939  * const struct netbuf *addr; -- netconfig structure
940  */
942 /*
943  * Generic TLI create routine
944 */
945 extern SVCXPRT *svc_tli_create(const int, const struct netconfig *,
946                                 const struct t_bind *, const uint_t,
947                                 const uint_t);
```

2

```

948     /*
949      *      const int fd;                      -- connection end point
950      *      const struct netconfig *nconf;    -- netconfig structure
951      *      const struct t_bind *bindaddr;   -- local bind address
952      *      const uint_t sendsz;           -- max sendsize
953      *      const uint_t recvsz;           -- max recvsize
954     */
955
956 /**
957  * Connectionless and connectionful create routines.
958 */
959 extern SVCXPRT *svc_vc_create(const int, const uint_t, const uint_t);
960 /*
961  *      const int fd;                      -- open connection end point
962  *      const uint_t sendsize;           -- max send size
963  *      const uint_t recvsz;           -- max recv size
964 */
965
966 extern SVCXPRT *svc_dg_create(const int, const uint_t, const uint_t);
967 /*
968  *      const int fd;                      -- open connection
969  *      const uint_t sendsize;           -- max send size
970  *      const uint_t recvsz;           -- max recv size
971 */
972
973 /**
974  * the routine takes any *open* TLI file
975  * descriptor as its first input and is used for open connections.
976 */
977 extern SVCXPRT *svc_fd_create(const int, const uint_t, const uint_t);
978 /*
979  *      const int fd;                      -- open connection end point
980  *      const uint_t sendsize;           -- max send size
981  *      const uint_t recvsz;           -- max recv size
982 */
983
984 /**
985  * Memory based rpc (for speed check and testing)
986 */
987 extern SVCXPRT *svc_raw_create(void);
988
989 /**
990  * Creation of service over doors transport.
991 */
992 extern SVCXPRT *svc_door_create(void (*)(struct svc_req *, SVCXPRT *),
993                                const rpcprog_t, const rpcvers_t,
994                                const uint_t);
995 /*
996  *      void (*dispatch)();                -- dispatch routine
997  *      const rpcprog_t program;          -- program number
998  *      const rpcvers_t versnum;         -- version number
999  *      const uint_t sendsize;           -- send buffer size
1000 */
1001
1002 /**
1003  * Service control interface
1004 */
1005 extern bool_t svc_control(SVCXPRT *, const uint_t, void *);
1006 /*
1007  *      SVCXPRT *svc;                  -- service to manipulate
1008  *      const uint_t req;              -- request
1009  *      void *info;                 -- argument to request
1010 */
1011
1012 /**
1013  * svc_dg_enable_cache() enables the cache on dg transports.

```

```

1014 /*
1015 extern int svc_dg_enablecache(SVCXPRT *, const uint_t);
1016 #else /* __STDC__ */
1017 extern int svc_create();
1018 extern SVCXPRT *svc_tp_create();
1019 extern SVCXPRT *svc_tli_create();
1020 extern SVCXPRT *svc_vc_create();
1021 extern SVCXPRT *svc_dg_create();
1022 extern SVCXPRT *svc_fd_create();
1023 extern SVCXPRT *svc_raw_create();
1024 extern SVCXPRT *svc_door_create();
1025 extern int svc_dg_enablecache();
1026#endif /* __STDC__ */
1027
1028 extern boolean_t is_multilevel(rpcprog_t);
1029
1030 #ifdef PORTMAP
1031 /* For backward compatibility */
1032 #include <rpc/svc_soc.h>
1033#endif /* PORTMAP */
1034
1035 /*
1036  * For user level MT hot server functions
1037 */
1038
1039 /*
1040  * Different MT modes
1041 */
1042 #define RPC_SVC_MT_NONE 0      /* default, single-threaded */
1043 #define RPC_SVC_MT_AUTO 1      /* automatic MT mode */
1044 #define RPC_SVC_MT_USER 2      /* user MT mode */
1045
1046 #ifdef __STDC__
1047 extern void svc_done(SVCXPRT *);
1048#else
1049 extern void svc_done();
1050#endif /* __STDC__ */
1051
1052 /*
1053  * Obtaining local credentials.
1054 */
1055 typedef struct __svc_local_cred_t {
1056     uid_t euid; /* effective uid */
1057     gid_t egid; /* effective gid */
1058     uid_t ruid; /* real uid */
1059     gid_t rgid; /* real gid */
1060     pid_t pid; /* caller's pid, or -1 if not available */
1061 } svc_local_cred_t;
1062
1063 unchanged_portion_omitted

```