

new/exception_lists/packaging

1

```

*****
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 #
21 #
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 #
30 #
31 #
32 # Exception List for validate_pkg
33 #
34 #
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 #

```

new/exception_lists/packaging

2

```

59 lib/libkcf.so
60 lib/libelfsign
61 lib/libelfsign.ln
62 lib/liblkcfd
63 lib/liblkcfd.ln
64 usr/include/libelfsign.h
65 usr/lib/liblsoftcrypto
66 usr/lib/liblsoftcrypto.ln
67 usr/lib/amd64/liblsoftcrypto.ln      i386
68 usr/lib/sparcv9/liblsoftcrypto.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/libdlether.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/libipadm
120 lib/libipadm.ln
121 lib/libipadm.so
122 #
123 # Private libsocket header file
124 #

```

new/exception_lists/packaging

3

```

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 # libipseutil. 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/libipseutil.so
138 usr/lib/amd64/libipseutil.so    i386
139 usr/lib/sparcv9/libipseutil.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-lipseutil
145 usr/lib/llib-lipseutil.ln
146 usr/lib/amd64/llib-lipseutil.ln i386
147 usr/lib/sparcv9/llib-lipseutil.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/libnwam_priv.h
158 usr/include/protocols/ripngd.h
159 usr/include/s_string.h
160 usr/include/sys/logindmux_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/ipsecesp.h
169 usr/include/inet/keysock.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 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

```

new/exception_lists/packaging

4

```

191 usr/lib/llib-lldisasm
192 usr/lib/llib-lldisasm.ln
193 usr/lib/amd64/llib-lldisasm.ln i386
194 usr/lib/sparcv9/llib-lldisasm.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-lraidcfg
204 usr/lib/llib-lraidcfg.ln
205 usr/lib/amd64/llib-lraidcfg.ln i386
206 usr/lib/sparcv9/llib-lraidcfg.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 libdhcpage
215 # and libdhcputil for the benefit of DHCP-related networking commands such
216 # as dhcpage, dhcpageinfo, ifconfig, and netstat. These are not interfaces
217 # for customer use, so the files are excluded from packaging.
218 #
219 lib/libdhcpage.so
220 lib/libdhcputil.so
221 lib/amd64/libdhcputil.so       i386
222 lib/sparcv9/libdhcputil.so     sparc
223 lib/llib-ldhcpage
224 lib/llib-ldhcpage.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/dhcpage_ipc.h
235 usr/include/dhcpage_util.h
236 usr/include/dhcpmsg.h
237 usr/lib/libdhcpage.so
238 usr/lib/libdhcputil.so
239 usr/lib/amd64/libdhcputil.so   i386
240 usr/lib/sparcv9/libdhcputil.so sparc
241 usr/lib/llib-ldhcpage
242 usr/lib/llib-ldhcpage.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

```

new/exception_lists/packaging

```

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/l1ib-linetsvc
269 usr/lib/l1ib-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/l1ib-linetutil
279 lib/l1ib-linetutil.ln
280 lib/amd64/l1ib-linetutil.ln      i386
281 lib/sparcv9/l1ib-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/l1ib-linetutil
289 usr/lib/l1ib-linetutil.ln
290 usr/lib/amd64/l1ib-linetutil.ln  i386
291 usr/lib/sparcv9/l1ib-linetutil.ln sparc
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

```

5

new/exception_lists/packaging

```

323 #
324 # these are "removed" from the source product build because the only
325 # packages that currently deliver them are removed.
326 # they really shouldn'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 ecode project which user does not need to see
342 #
343 usr/platform/sun4u/include/sys/fc_plat.h          sparc
344 usr/platform/sun4u/include/sys/fcode.h           sparc
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 # Simmet
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          i386
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/l1ib-lrestart
385 lib/l1ib-lrestart.ln
386 lib/amd64/l1ib-lrestart.ln          i386
387 lib/sparcv9/l1ib-lrestart.ln       sparc
388 usr/include/libcontract_priv.h

```

6

new/exception_lists/packaging

7

```

389 usr/include/librestart_priv.h
390 usr/include/librestart.h
391 usr/lib/librestart.so
392 usr/lib/sparcv9/librestart.so          sparc
393 lib/svc/manifest/platform/sun4u       i386
394 lib/svc/manifest/platform/sun4v       i386
395 var/svc/manifest/platform/sun4u       i386
396 var/svc/manifest/platform/sun4v       i386
397 etc/svc/profile/platform_sun4v.xml    i386
398 etc/svc/profile/platform_SUNW,SPARC-Enterprise.xml i386
399 etc/svc/profile/platform_SUNW,Sun-Fire-15000.xml i386
400 etc/svc/profile/platform_SUNW,Sun-Fire-880.xml i386
401 etc/svc/profile/platform_SUNW,Sun-Fire-V890.xml i386
402 etc/svc/profile/platform_SUNW,Sun-Fire.xml i386
403 etc/svc/profile/platform_SUNW,Ultra-Enterprise-10000.xml i386
404 etc/svc/profile/platform_SUNW,UltraSPARC-IIe-NetraCT-40.xml i386
405 etc/svc/profile/platform_SUNW,UltraSPARC-IIe-NetraCT-60.xml i386
406 etc/svc/profile/platform_SUNW,UltraSPARC-IIi-Netract.xml i386
407 #
408 # Private libuutil files
409 #
410 lib/libuutil.so
411 lib/l1ib-luutil
412 lib/l1ib-luutil.ln
413 lib/sparcv9/l1ib-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/l1ib-lcmdutils
452 lib/l1ib-lcmdutils.ln
453 lib/amd64/l1ib-lcmdutils.ln           i386
454 lib/sparcv9/l1ib-lcmdutils.ln         sparc

```

new/exception_lists/packaging

8

```

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/l1ib-lcmdutils
460 usr/lib/l1ib-lcmdutils.ln
461 usr/lib/amd64/l1ib-lcmdutils.ln       i386
462 usr/lib/sparcv9/l1ib-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/l1ib-lprtdiag    sparc
485 usr/platform/sun4u/lib/l1ib-lprtdiag.ln sparc
486 usr/platform/sun4v/lib/l1ib-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                 sparc
493 usr/include/sys/mdesc_impl.h            sparc
494 usr/platform/sun4v/include/sys/mach_descrip.h sparc
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/l1ib-lpcp.ln     sparc
501 usr/platform/SUNW,Netra-CP3060/lib/l1ib-lpcp.ln sparc
502 usr/platform/SUNW,Netra-CP3260/lib/l1ib-lpcp.ln sparc
503 usr/platform/SUNW,Netra-T5220/lib/l1ib-lpcp.ln sparc
504 usr/platform/SUNW,Netra-T5440/lib/l1ib-lpcp.ln sparc
505 usr/platform/SUNW,SPARC-Enterprise-T5120/lib/l1ib-lpcp.ln sparc
506 usr/platform/SUNW,Sun-Blade-T6300/lib/l1ib-lpcp.ln sparc
507 usr/platform/SUNW,Sun-Blade-T6320/lib/l1ib-lpcp.ln sparc
508 usr/platform/SUNW,Sun-Fire-T200/lib/l1ib-lpcp.ln sparc
509 usr/platform/SUNW,T5140/lib/l1ib-lpcp.ln sparc
510 usr/platform/SUNW,USBDRD-5240/lib/l1ib-lpcp.ln sparc
511 #
512 # ZFS internal tools and lint libraries
513 #
514 usr/lib/l1ib-lzfs_jni
515 usr/lib/l1ib-lzfs_jni.ln
516 usr/lib/amd64/l1ib-lzfs_jni.ln         i386
517 usr/lib/sparcv9/l1ib-lzfs_jni.ln       sparc
518 usr/lib/l1ib-lzpool
519 usr/lib/l1ib-lzpool.ln                  i386
520 usr/lib/amd64/l1ib-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_diskmgmt.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/isns_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_autofs.so
627 usr/lib/fs/autofs/amd64/libshare_autofs.so      i386
628 usr/lib/fs/autofs/sparcv9/libshare_autofs.so    sparc
629 usr/lib/fs/nfs/libshare_nfs.so
630 usr/lib/fs/nfs/amd64/libshare_nfs.so            i386
631 usr/lib/fs/nfs/sparcv9/libshare_nfs.so         sparc
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            i386
635 usr/lib/fs/smb/sparcv9/libshare_smb.so         sparc
636 usr/lib/fs/smbfs/libshare_smbfs.so
637 usr/lib/fs/smbfs/amd64/libshare_smbfs.so       i386
638 usr/lib/fs/smbfs/sparcv9/libshare_smbfs.so     sparc
639 usr/include/libshare_impl.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                  sparc
649 usr/lib/llib-lpri                 sparc
650 usr/lib/llib-lpri.ln              sparc
651 usr/lib/sparcv9/libpri.so          sparc
652 usr/lib/sparcv9/llib-lpri.ln     sparc

```

```

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/llib-lds               sparc
663 usr/lib/llib-lds.ln           sparc
664 usr/lib/sparcv9/llib-lds.ln    sparc
665 usr/lib/libdscfg.so
666 usr/lib/llib-ldscfg.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/libsqlite-native.o
678 lib/libsqlite-sys.so
679 lib/llib-lsqlite-sys
680 lib/llib-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_ck_common.h
686 usr/include/sys/kiconv_cp950hkscs_utf8.h
687 usr/include/sys/kiconv_emeal.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 #

```

```

719 usr/include/sys/srn.h
720 #
721 # Private/Internal header files of smbdrv. Do not ship.
722 #
723 usr/include/smb
724 usr/include/smbdrv
725 #
726 # Private/Internal libraries for smbdrv in user space
727 #
728 usr/include/libfakekernel
729 usr/lib/libfakekernel.so
730 usr/lib/libfakekernel.so.1
731 usr/lib/llib-lfakekernel
732 usr/lib/llib-lfakekernel.ln
733 usr/lib/amd64/llib-lfakekernel.ln      i386
734 usr/lib/sparcv9/llib-lfakekernel.ln    sparc
735 usr/lib/amd64/libfakekernel.so         i386
736 usr/lib/amd64/libfakekernel.so.1       i386
737 usr/lib/sparcv9/libfakekernel.so        sparc
738 usr/lib/sparcv9/libfakekernel.so.1     sparc
739 #
740 # Private/Internal libraries of smbdrv. Do not ship.
741 #
742 usr/lib/mdb/proc/libfksmbdrv.so
743 usr/lib/mdb/proc/amd64/libfksmbdrv.so   i386
744 usr/lib/mdb/proc/sparcv9/libfksmbdrv.so sparc
745 usr/lib/reparse/llib-lreparse_smb
746 usr/lib/reparse/llib-lreparse_smb.ln
747 usr/lib/smbdrv/bind-helper
748 usr/lib/smbdrv/fksmbd
749 usr/lib/smbdrv/libfksmbdrv.so
750 usr/lib/smbdrv/libfksmbdrv.so.1
751 usr/lib/smbdrv/llib-lfksmbdrv
752 usr/lib/smbdrv/llib-lfksmbdrv.ln
753 usr/lib/smbdrv/llib-lmlrpc
754 usr/lib/smbdrv/llib-lmlrpc.ln
755 usr/lib/smbdrv/llib-lmlsvc
756 usr/lib/smbdrv/llib-lmlsvc.ln
757 usr/lib/smbdrv/llib-lsmb
758 usr/lib/smbdrv/llib-lsmb.ln
759 usr/lib/smbdrv/llib-lsmbns
760 usr/lib/smbdrv/llib-lsmbns.ln
761 #
762 #
763 # Private/Internal 64-bit libraries of smbdrv. Do not ship.
764 #
765 usr/lib/smbdrv/amd64              i386
766 usr/lib/smbdrv/sparcv9            sparc

768 usr/lib/reparse/amd64/libreparse_smb.so  i386
769 usr/lib/reparse/amd64/libreparse_smb.so.1 i386
770 usr/lib/reparse/amd64/llib-lreparse_smb.ln i386
771 usr/lib/reparse/sparcv9/libreparse_smb.so sparc
772 usr/lib/reparse/sparcv9/libreparse_smb.so.1 sparc
773 usr/lib/reparse/sparcv9/llib-lreparse_smb.ln sparc
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 vsfan service
780 #
781 usr/include/libvscan.h
782 usr/include/sys/vscan.h
783 #
784 # libvscan is private

```

new/exception_lists/packaging

13

```

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 # libsmbfs is private
800 #
801 usr/include/net smb
802 usr/lib/libsmbfs.so
803 usr/lib/amd64/libsmbfs.so           i386
804 usr/lib/sparcv9/libsmbfs.so       sparc
805 usr/lib/llib-lsmbfs
806 usr/lib/llib-lsmbfs.ln
807 usr/lib/amd64/llib-lsmbfs.ln     i386
808 usr/lib/sparcv9/llib-lsmbfs.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-LHBAAPI
847 usr/lib/llib-LHBAAPI.ln
848 usr/lib/amd64/llib-LHBAAPI.ln  i386
849 usr/lib/sparcv9/llib-LHBAAPI.ln sparc
850 #

```

new/exception_lists/packaging

14

```

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/librdc.so
863 usr/lib/libunistat.so
864 usr/lib/llib-lnsctl.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

```

new/exception_lists/packaging

15

```

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/l1ib-lvrrpadm
925 usr/lib/l1lib-lvrrpadm.ln
926 usr/lib/amd64/l1lib-lvrrpadm.ln    i386
927 usr/lib/sparcv9/l1lib-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/libstandssaveargs.so i386
967 usr/lib/amd64/l1lib-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/l1lib-lpcidb.ln     i386
975 usr/lib/sparcv9/libpcidb.so       sparc
976 usr/lib/sparcv9/l1lib-lpcidb.ln   sparc
977 usr/lib/libpcidb.so
978 usr/lib/l1lib-lpcidb
979 usr/lib/l1lib-lpcidb.ln

981 #
982 # private nvme header file

```

new/exception_lists/packaging

16

```

983 #
984 usr/include/sys/nvme.h

986 #
987 # debugging program for libadutils
988 #
989 usr/bin/test-getdc
990 #
991 # libficl-sys is private
992 #
993 usr/include/ficllocal.h
994 usr/lib/amd64/l1lib-lficl-sys.ln   i386
995 usr/lib/amd64/libficl-sys.so       i386
996 usr/lib/sparcv9/l1lib-lficl-sys.ln sparc
997 usr/lib/sparcv9/libficl-sys.so     sparc
998 usr/lib/l1lib-lficl-sys
999 usr/lib/l1lib-lficl-sys.ln
1000 usr/lib/libficl-sys.so

```


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

1

```
*****
1807 Mon Jul 17 16:15:15 2017
new/usr/src/cmd/fs.d/nfs/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 (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 #
31 #
32 include $(SRC)/Makefile.master
33 #
34 SUBDIR1=      exportfs nfsd rquotad \
35              statd nfsstat mountd dfshares \
36              nfsfind nfs4cbd share tests
37              nfsfind nfs4cbd share
38 # These do "make catalog"
39 SUBDIR2=      clear_locks lockd umount showmount \
40              mount dfmounts nfslog nfsmapid \
41              nfsref rp_basic
42 #
43 SUBDIR3=      etc svc
44 SUBDIRS=      $(SUBDIR1) $(SUBDIR2) $(SUBDIR3)
45 #
46 # for messaging catalog files
47 #
48 POFILES=     $(SUBDIR2:%=%/.po)
49 POFILE=      nfs.po
50 #
51 all:=        TARGET= all
52 install:=    TARGET= install
53 clean:=      TARGET= clean
54 clobber:=    TARGET= clobber
55 lint:=       TARGET= lint
56 catalog:=    TARGET= catalog
```

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

2

```
58 .KEEP_STATE:
59 #
60 .PARALLEL:   $(SUBDIRS)
61 #
62 all install clean clobber lint: $(SUBDIRS)
63 #
64 catalog:     $(SUBDIR2)
65              $(RM) $(POFILE)
66              cat $(POFILES) > $(POFILE)
67 #
68 $(SUBDIRS): FRC
69              @cd $@; pwd; $(MAKE) $(TARGET)
70 #
71 FRC:
```

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

1

```
*****
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 */
21
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  * Copyright 2016 Nexenta Systems, Inc. All rights reserved.
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 <rpcsvc/mount.h>
52 #include <sys/pathconf.h>
53 #include <sys/systeminfo.h>
54 #include <sys/utname.h>
55 #include <sys/wait.h>
56 #include <sys/resource.h>
57 #include <signal.h>
```

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

2

```
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 <rpcsvc/daemon_utils.h>
73 #include <deflt.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>
91
92 extern int daemonize_init(void);
93 extern void daemonize_fini(int);
94
95 extern int _nfssys(int, void *);
96
97 struct sh_list *share_list;
98
99 rwlock_t sharetab_lock;      /* lock to protect the cached sharetab */
100 static mutex_t mnttab_lock; /* prevent concurrent mnttab readers */
101
102 static mutex_t logging_queue_lock;
103 static cond_t logging_queue_cv;
104
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 *r);
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 *);
120
121 static int debug;
122 static int verbose;
123 static int rejecting;
```

```

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 *xprt = NULL;
383     rpcvers_t vers;

385     vers = mount_vers_max;

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_protofmly, NC_INET) == 0 ||
396          strcmp(nconf->nc_protofmly, NC_INET6) == 0)) {
397         int err;

399         snprintf(port_str, sizeof (port_str), "%u",
400                 (unsigned short)mountd_port);

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             xprt = svc_tp_create_addr(mnt, MOUNTPROG, vers,
407                                     nconf, al->n_addrs);
408             netdir_free(al, ND_ADDRLIST);
409         }
410         if (xprt == 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 (xprt == 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     xprt = svc_tp_create(mnt, MOUNTPROG, vers, nconf);
424 }
425 if (xprt == NULL) {
426     syslog(LOG_ERR, "mountd: unable to create "
427            "(MOUNTD,%d) on transport %s",
428            vers, nconf->nc_netid);
429     return;
430 }

432 /*
433 * Register additional versions on this transport.
434 */
435 while (--vers >= mount_vers_min) {
436     if (!svc_reg(xprt, 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 }

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

464     int pipe_fd = -1;

466     /*
467     * Mountd requires uid 0 for:
468     * /etc/rmtab updates (we could chown it to daemon)
469     * /etc/dfs/dfstab reading (it wants to lock out share which
470     * doesn't do any locking before first truncate;
471     * NFS share does; should use fcntl locking instead)
472     * Needed privileges:
473     * auditing
474     * nfs syscall
475     * file dac search (so it can stat all files)
476     * Optional privileges:
477     * MLP
478     */

```

```

479     can_do_mlp = priv_ineffect(PRIV_NET_BINDMLP);
480     if (!__init_daemon_priv(PU_RESETPROGS|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     }

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     }

498     (void) enable_extended_FILE_stdio(-1, -1);

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     }

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     }

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 }

549 /*
550  * Read in the NFS version values from config file.
551  */
552     bufksz = 4;
553     ret = nfs_smf_get_prop("server_versmin", defval, DEFAULT_INSTANCE,
554         SCF_TYPE_INTEGER, NFSD, &bufksz);
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     }

569     bufksz = 4;
570     ret = nfs_smf_get_prop("server_versmax", defval, DEFAULT_INSTANCE,
571         SCF_TYPE_INTEGER, NFSD, &bufksz);
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     }

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     }

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);

604     netgroup_init();

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 (mountd_port < 0 || mountd_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     }

745     /*
746     * Create an additional thread to service the rmtab and
747     * audit_mountd_mount logging for mount requests. Use the same
748     * signal disposition as the main thread. We create
749     * a separate thread to allow the mount request threads to
750     * clear as soon as possible.
751     */
752     if (thr_create(NULL, 0, logging_svc, 0, thr_flags, &logging_thread)) {
753         syslog(LOG_ERR, gettext("Failed to create LOGGING svc thread"));
754         exit(2);
755     }

757     /*
758     * Enumerate network transports and create service listeners
759     * as appropriate for each.
760     * Create datagram and connection oriented services
761     */
762     if ((nc = setnetconfig()) == NULL) {
763         syslog(LOG_ERR, "setnetconfig failed: %m");
764         return (-1);
765     }
766     if (mount_vers_max >= MOUNTVERS) {
767         if (svc_create(mnt, MOUNTPROG, MOUNTVERS, "datagram_v") == 0) {
768             fprintf(stderr,
769                 "couldn't register datagram_v MOUNTVERS\n");
770             exit(1);
771         }
772     }
773     while ((nconf = getnetconfig(nc)) != NULL) {
774         /*
775         * Skip things like tpi_raw, invisible...
776         */
777         if ((nconf->nc_flag & NC_VISIBLE) == 0)
778             continue;
779         if (nconf->nc_semantics != NC_TPI_CLTS &&
780             nconf->nc_semantics != NC_TPI_COTS &&
781             nconf->nc_semantics != NC_TPI_COTS_ORD)
782             continue;
783         if (svc_create(mnt, MOUNTPROG, MOUNTVERS, "circuit_v") == 0) {
784             fprintf(stderr,
785                 "couldn't register circuit_v MOUNTVERS\n");
786             exit(1);
787         }
788     }

789     md_svc_tp_create(nconf);
790     if (mount_vers_max >= MOUNTVERS_POSIX) {
791         if (svc_create(mnt, MOUNTPROG, MOUNTVERS_POSIX,
792             "datagram_v") == 0) {
793             fprintf(stderr,
794                 "couldn't register datagram_v MOUNTVERS_POSIX\n");
795             exit(1);
796         }
797     }
798     (void) endnetconfig(nc);
799     if (svc_create(mnt, MOUNTPROG, MOUNTVERS_POSIX,
800         "circuit_v") == 0) {
801         fprintf(stderr,
802             "couldn't register circuit_v MOUNTVERS_POSIX\n");
803         exit(1);
804     }
805 }

807 if (mount_vers_max >= MOUNTVERS3) {
808     if (svc_create(mnt, MOUNTPROG, MOUNTVERS3, "datagram_v") == 0) {
809         fprintf(stderr,

```

```

810         "couldn't register datagram_v MOUNTVERS3\n");
811         exit(1);
812     }
813     if (svc_create(mnt, MOUNTPROG, MOUNTVERS3, "circuit_v") == 0) {
814         fprintf(stderr,
815             "couldn't register circuit_v MOUNTVERS3\n");
816         exit(1);
817     }
818 }

820 /*
821 * Start serving
822 */
823 rmtab_load();

825 daemonize_fini(pipe_fd);

827 /* Get rid of the most dangerous basic privileges. */
828 __fini_daemon_priv(PRIV_PROC_EXEC, PRIV_PROC_INFO, PRIV_PROC_SESSION,
829     (char *)NULL);

831 svc_run();
832 syslog(LOG_ERR, "Error: svc_run shouldn't have returned");
833 abort();

835 /* NOTREACHED */
836 return (0);
837 }

```

unchanged_portion_omitted

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

1

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 ATMK=        $(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 OBJS=        $(LOCAL) selfcheck.o daemon.o smfcfg.o
44 OBJS=        $(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): $(OBJS)
```

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

2

```
56 $(LINK.c) -o $@ $(OBJS) $(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) $(OBJS) $(TYPEPROG)
```

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

1

```
*****
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 */
21
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  * Copyright 2016 Nexenta Systems, Inc. All rights reserved.
27 */
28 /*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
29 /*      All Rights Reserved      */
30
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 /*
42  * Copyright (c) 2012 by Delphix. All rights reserved.
43  */
44 #include <stdio.h>
45 #include <stdio_ext.h>
46 #include <stdlib.h>
47 #include <ftw.h>
48 #include <signal.h>
49 #include <string.h>
50 #include <syslog.h>
51 #include <netconfig.h>
52 #include <netdir.h>
53 #include <unistd.h>
54 #include <syslog.h>
55 #include <netdb.h>
56 #include <rpc/rpc.h>
57 #include <rpc/svc.h>
```

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

2

```
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 <rpcsvc/sm_inter.h>
64 #include <rpcsvc/nsm_addr.h>
65 #include <thread.h>
66 #include <synch.h>
67 #include <net/if.h>
68 #include <limits.h>
69 #include <rpcsvc/daemon_utils.h>
70 #include <priv_utils.h>
71 #include "smcfg.h"
72 #include "sm_statd.h"
73
74 #define home0      "/var/statmon"
75 #define current0   "/var/statmon/sm"
76 #define backup0   "/var/statmon/sm.bak"
77 #define state0    "/var/statmon/state"
78
79 #define homel      "statmon"
80 #define current1   "statmon/sm/"
81 #define backup1   "statmon/sm.bak/"
82 #define statel    "statmon/state"
83
84 extern int  daemonize_init(void);
85 extern void daemonize_fini(int fd);
86
87
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;
96
97 char STATE[MAXPATHLEN], CURRENT[MAXPATHLEN], BACKUP[MAXPATHLEN];
98 static char statd_home[MAXPATHLEN];
99
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];
104
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  addrix; /* # of -a entries */
113
114
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.
120  */
```



```

252         xdr_argument = xdr_mon;
253         xdr_result = xdr_sm_stat_res;
254         local = (char *(*()) sm_mon_svc;
255         break;

257     case SM_UNMON:
258         xdr_argument = xdr_mon_id;
259         xdr_result = xdr_sm_stat;
260         local = (char *(*()) sm_unmon_svc;
261         break;

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;

269     case SM_SIMU_CRASH:
270         xdr_argument = xdr_void;
271         xdr_result = xdr_void;
272         local = (char *(*()) sm_simu_crash_svc;
273         break;

275     case SM_NOTIFY:
276         xdr_argument = xdr_stat_chge;
277         xdr_result = xdr_void;
278         local = (char *(*()) sm_notify_svc;
279         break;

281     default:
282         svcerr_noproc(transp);
283         return;
284     }
285 }

287 (void) memset(&argument, 0, sizeof (argument));
288 if (!svc_getargs(transp, xdr_argument, (caddr_t)&argument)) {
289     svcerr_decode(transp);
290     return;
291 }

293 (void) memset(&result, 0, sizeof (result));
294 (*local)(amp;argument, amp;result);
295 if (!svc_sendreply(transp, xdr_result, (caddr_t)&result)) {
296     svcerr_systemerr(transp);
297 }

299 if (!svc_freeargs(transp, xdr_argument, (caddr_t)&argument)) {
300     syslog(LOG_ERR, "statd: unable to free arguments\n");
301 }
302 }

    unchanged_portion_omitted

461 /*
462  * This function is called for each configured network type to
463  * bind and register our RPC service programs.
464  *
465  * On TCP or UDP, we may want to bind SM_PROG on a specific port
466  * (when statd_port is specified) in which case we'll use the
467  * variant of svc_tp_create() that lets us pass a bind address.
468  */
469 static void
470 sm_svc_tp_create(struct netconfig *nconf)
471 {
472     char port_str[8];
473     struct nd_hostserv hs;

```

```

474     struct nd_addrlist *al = NULL;
475     SVCXPRT *xpirt = NULL;

477     /*
478     * If statd_port is set and this is an inet transport,
479     * bind this service on the specified port.  The TLI way
480     * to create such a bind address is netdir_getbyname()
481     * with the special "host" HOST_SELF_BIND.  This builds
482     * an all-zeros IP address with the specified port.
483     */
484     if (statd_port != 0 &&
485         (strcmp(nconf->nc_protofmly, NC_INET) == 0 ||
486          strcmp(nconf->nc_protofmly, NC_INET6) == 0)) {
487         int err;

489         snprintf(port_str, sizeof (port_str), "%u",
490                 (unsigned short)statd_port);

492         hs.h_host = HOST_SELF_BIND;
493         hs.h_serv = port_str;
494         err = netdir_getbyname((struct netconfig *)nconf, &hs, &al);
495         if (err == 0 && al != NULL) {
496             xpirt = svc_tp_create_addr(sm_prog_1, SM_PROG, SM_VERS,
497                                     nconf, al->n_addrs);
498             netdir_free(al, ND_ADDRLIST);
499         }
500         if (xpirt == NULL) {
501             syslog(LOG_ERR, "statd: unable to create "
502                  "(SM_PROG, SM_VERS) on transport %s (port %d)",
503                  nconf->nc_netid, statd_port);
504         }
505         /* fall-back to default bind */
506     }
507     if (xpirt == NULL) {
508         /*
509         * Had statd_port=0, or non-inet transport,
510         * or the bind to a specific port failed.
511         * Do a default bind.
512         */
513         xpirt = svc_tp_create(sm_prog_1, SM_PROG, SM_VERS, nconf);
514     }
515     if (xpirt == NULL) {
516         syslog(LOG_ERR, "statd: unable to create "
517              "(SM_PROG, SM_VERS) for transport %s",
518              nconf->nc_netid);
519         return;
520     }

522     /*
523     * Also register the NSM_ADDR program on this
524     * transport handle (same dispatch function).
525     */
526     if (!svc_reg(xpirt, NSM_ADDR_PROGRAM, NSM_ADDR_V1, sm_prog_1, nconf)) {
527         syslog(LOG_ERR, "statd: failed to register "
528              "(NSM_ADDR_PROGRAM, NSM_ADDR_V1) for "
529              "netconfig %s", nconf->nc_netid);
530     }
531 }

533 int
534 main(int argc, char *argv[])
535 {
536     int c;
537     int ppid;
538     extern char *optarg;
539     int choice = 0;

```

```

540     struct rlimit rl;
541     int mode;
542     int sz;
543     int pipe_fd = -1;
544     int ret;
545     int connmaxrec = 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: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                     } else
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             break;
592         }
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

```

```

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         if (path_name == NULL) {
611             size_t sz = (argc/2) * sizeof(char *);

613             path_name = (char **)malloc(sz);
614             if (path_name == NULL) {
615                 (void) fprintf(stderr,
616                     "statd: malloc failed\n");
617                 exit(1);
618             }
619             (void) memset(path_name, 0, sz);
620         }
621         path_name[pathix] = optarg;
622         pathix++;
623     } else {
624         (void) fprintf(stderr,
625             "statd: -p pathname is too long.\n");
626     }
627     break;
628 case 'P':
629     (void) sscanf(optarg, "%d", &statd_port);
630     if (statd_port < 1 || statd_port > UINT16_MAX) {
631         (void) fprintf(stderr,
632             "statd: -P port invalid.\n");
633         statd_port = 0;
634     }
635     break;
636 case 'r':
637     regfiles_only = 1;
638     break;
639 default:
640     (void) fprintf(stderr,
641         "statd [-d level] [-D]\n");
642     return (1);
643 }

645 if (choice == 0) {
646     (void) strcpy(statd_home, home0);
647     (void) strcpy(CURRENT, current0);
648     (void) strcpy(BACKUP, backup0);
649     (void) strcpy(STATE, state0);
650 } else {
651     (void) strcpy(statd_home, homel);
652     (void) strcpy(CURRENT, current1);
653     (void) strcpy(BACKUP, backup1);
654     (void) strcpy(STATE, statel);
655 }
656 if (debug)
657     (void) printf("debug is on, create entry: %s, %s, %s\n",
658         CURRENT, BACKUP, STATE);

660 if (getrlimit(RLIMIT_NOFILE, &rl))
661     (void) printf("statd: getrlimit failed. \n");

663 /* Set maxfdlimit current soft limit */
664 rl.rlim_cur = rl.rlim_max;
665 if (setrlimit(RLIMIT_NOFILE, &rl) != 0)
666     syslog(LOG_ERR, "statd: unable to set RLIMIT_NOFILE to %d\n",
667         rl.rlim_cur);

669 (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", STATD,
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          * Skip things like tpi_raw, invisible...
647          */
648         if ((nconf->nc_flag & NC_VISIBLE) == 0)
649             continue;
650         if (nconf->nc_semantics != NC_TPI_CLTS &&
651             nconf->nc_semantics != NC_TPI_COTS &&
652             nconf->nc_semantics != NC_TPI_COTS_ORD)
653             continue;
654         sm_svc_tp_create(nconf);
655     if (!svc_create(sm_prog_1, NSM_ADDR_PROGRAM, NSM_ADDR_V1, "netpath")) {
656         syslog(LOG_ERR, "statd: unable to create (NSM_ADDR_PROGRAM, "
657               "NSM_ADDR_V1) for netpath.");
658     }
659     (void) endnetconfig(nc);
660
661     /*
662     * Make sure /var/statmon and any alternate (-p) statmon
663     * directories exist and are owned by daemon. Then change our uid
664     * to daemon. The uid change is to prevent attacks against local
665     * daemons that trust any call from a local root process.
666     */
667     set_statmon_owner();
668
669     /*
670     *
671     * statd now runs as a daemon rather than root and can not
672     * dump core under / because of the permission. It is
673     * important that current working directory of statd be
674     * changed to writable directory /var/statmon so that it
675     * can dump the core upon the receipt of the signal.
676     * One still need to set allow_setid_core to non-zero in
677     * /etc/system to get the core dump.
678     */
679     if (chdir(statd_home) < 0) {
680         syslog(LOG_ERR, "can't chdir %s: %m", statd_home);
681         exit(1);
682     }
683
684     copy_client_names();
685
686     rwlock_init(&thr_rwlock, USYNC_THREAD, NULL);
687     mutex_init(&crash_lock, USYNC_THREAD, NULL);
688     mutex_init(&name_addrlock, USYNC_THREAD, NULL);
689     cond_init(&retrywait, USYNC_THREAD, NULL);
690     sm_inithash();
691     die = 0;
692     /*
693     * This variable is set to ensure that an sm_crash
694     * request will not be done at the same time
695     * when a statd_init is being done, since sm_crash
696     * can reset some variables that statd_init will be using.
697     */
698     in_crash = 1;
699     statd_init();

```

```
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 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 <?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 reser
25 Copyright (c) 2012 by Delphix. All rights reserved.
26 Copyright 2014 Nexenta Systems, Inc. All rights reserved
27 Copyright 2016 Hens Rosenfeld <rosenfeld@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='SUNWnfsr: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' />

129 <exec_method
130     type='method'
131     name='refresh'
132     exec='/lib/svc/method/nfs-server %m'
133     timeout_seconds='3600' />

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' />

146 <property_group name='application' type='framework'>
147     <stability value='Evolving' />
148     <propval name='auto_enable' type='boolean' value='true' />
149 </property_group>

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>

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>

186 <stability value='Stable' />

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>

201 </service_bundle>

```

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

1

```
*****
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'
```

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

2

```
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>
```


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

1

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

1

```
*****
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 /*
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 2017 Nexenta Systems, Inc. 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 <stdlib.h>
38 #include <ctype.h>
39 #include <sys/types.h>
40 #include <string.h>
41 #include <syslog.h>
42 #include <sys/param.h>
43 #include <rpc/rpc.h>
44 #include <sys/stat.h>
45 #include <netconfig.h>
46 #include <netdir.h>
47
48 #include <sys/file.h>
49 #include <sys/time.h>
50 #include <sys/errno.h>
51 #include <rpcsvc/mount.h>
52
53 #include <signal.h>
54 #include <locale.h>
55 #include <unistd.h>
56 #include <errno.h>
57 #include <sys/socket.h>
58 #include <netinet/in.h>
```

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

2

```
59 #include <arpa/inet.h>
60 #include <netdb.h>
61
62 #include <thread.h>
63 #include <assert.h>
64
65 #include <limits.h>
66
67 #define TESTPROG      987654
68
69 uint32_t test_vers_max = 2;
70 uint32_t test_vers_min = 1;
71
72 int debug;
73 int verbose;
74 int testd_port;
75
76 static void mysvc(struct svc_req *, SVCXPRT *);
77 static void bind2(void);
78
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 *xpirt = NULL;
94     rpcvers_t vers;
95
96     vers = test_vers_max;
97
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_protofmly, NC_INET) == 0 ||
104          strcmp(nconf->nc_protofmly, NC_INET6) == 0)) {
105         int err;
106
107         snprintf(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             xpirt = svc_tp_create_addr(mysvc, TESTPROG, vers,
115                                       nconf, al->n_addrs);
116             netdir_free(al, ND_ADDRLIST);
117         }
118         if (xpirt == 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     }
125 }
```

```

125     if (xprt == NULL) {
126         /*
127          * Had testd_port=0, or non-inet transport,
128          * or the bind to a specific port failed.
129          * Do a default bind.
130          */
131         xprt = svc_tp_create(mysvc, TESTPROG, vers, nconf);
132     }
133     if (xprt == NULL) {
134         printf("testd: unable to create "
135              "(TESTD,%d) on transport %s\n",
136              vers, nconf->nc_netid);
137         return;
138     }
139
140     /*
141     * Register additional versions on this transport.
142     */
143     while (--vers >= test_vers_min) {
144         if (!svc_reg(xprt, TESTPROG, vers, mysvc, nconf)) {
145             printf("testd: "
146                  "failed to register vers %d on %s\n",
147                  vers, nconf->nc_netid);
148         }
149     }
150 }
151
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 }
160
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
170     while ((c = getopt(argc, argv, "dvp:")) != EOF) {
171         switch (c) {
172             case 'd':
173                 debug++;
174                 break;
175             case 'v':
176                 verbose++;
177                 break;
178             case 'p':
179                 (void) sscanf(optarg, "%d", &tmp);
180                 if (tmp < 1 || tmp > UINT16_MAX) {
181                     (void) fprintf(stderr,
182                                  "testd: -P port invalid.\n");
183                     return (1);
184                 }
185                 testd_port = tmp;
186                 break;
187             default:
188                 fprintf(stderr, "usage: testd [-v] [-r]\n");
189                 exit(1);
190         }

```

```

191     }
192
193     (void) setlocale(LC_ALL, "");
194
195     #if !defined(TEXT_DOMAIN)
196     #define TEXT_DOMAIN "SYS_TEST"
197     #endif
198     (void) textdomain(TEXT_DOMAIN);
199
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
208     if (testd_port < 0 || testd_port > UINT16_MAX) {
209         fprintf(stderr, "unable to use specified port\n");
210         exit(1);
211     }
212
213     /*
214     * Make sure to unregister any previous versions in case the
215     * user is reconfiguring the server in interesting ways.
216     */
217     test_svc_unreg();
218
219     /*
220     * Enumerate network transports and create service listeners
221     * as appropriate for each.
222     */
223     if ((nc = setnetconfig()) == NULL) {
224         perror("setnetconfig failed");
225         return (-1);
226     }
227     while ((nconf = getnetconfig(nc)) != NULL) {
228         /*
229          * Skip things like tpi_raw, invisible...
230          */
231         if ((nconf->nc_flag & NC_VISIBLE) == 0)
232             continue;
233         if (nconf->nc_semantics != NC_TPI_CLTS &&
234             nconf->nc_semantics != NC_TPI_COTS &&
235             nconf->nc_semantics != NC_TPI_COTS_ORD)
236             continue;
237
238         test_svc_tp_create(nconf);
239     }
240     (void) endnetconfig(nc);
241
242     /*
243     * XXX: Normally would call svc_run() here, but
244     * we just want to check our IP bindings.
245     */
246     if (testd_port != 0)
247         bind2();
248
249     if (debug) {
250         char sysbuf[100];
251
252         snprintf(sysbuf, sizeof(sysbuf),
253                  "rpcinfo -p |grep %u", TESTPROG);
254         printf("x %s\n", sysbuf);
255         fflush(stdout);
256         system(sysbuf);

```

```

258         if (testd_port) {
259             snprintf(sysbuf, sizeof (sysbuf),
260                 "netstat -a -f inet -P udp |grep %u", testd_por
261             );
262             printf("x %s\n", sysbuf);
263             fflush(stdout);
264             system(sysbuf);
265
266             snprintf(sysbuf, sizeof (sysbuf),
267                 "netstat -a -f inet -P tcp |grep %u", testd_por
268             );
269             printf("x %s\n", sysbuf);
270             fflush(stdout);
271             system(sysbuf);
272         }
273     }
274     /* cleanup */
275     test_svc_unreg();
276
277     printf("%s complete\n", argv[0]);
278     return (0);
279 }
280
281 /*
282  * Server procedure switch routine
283  */
284 static void
285 mysvc(struct svc_req *rq, SVCXPRT *xpirt)
286 {
287     switch (rq->rq_proc) {
288     case NULLPROC:
289         errno = 0;
290         (void)svc_sendreply(xpirt, xdr_void, (char *)0);
291         return;
292
293     default:
294         svcerr_noproc(xpirt);
295         return;
296     }
297 }
298
299 struct sockaddr_in addr;
300
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;
310
311     addr.sin_family = AF_INET;
312     addr.sin_port = htons(testd_port);
313     addr.sin_addr.s_addr = htonl(INADDR_LOOPBACK);
314
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     }
321
322     ret = bind(sock, (struct sockaddr *)&addr, sizeof (addr));

```

```

323         if (ret == -1) {
324             fprintf(stderr, "bind2 bind fail %s (expected) PASS\n", strerror
325             );
326             close(sock);
327             return;
328         }
329
330     printf("Oh no, bind2 worked! test FAILED\n");
331     close(sock);
332     return;
333 }

```

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 #
25 #
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 #
39 #
40 $mapfile_version 2
41 #
42 SYMBOL_VERSION ILLUMOS_0.1 { # Illumos additions
43     global:
44     svc_tp_create_addr;
45 } SUNW_1.10;
46 #
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

1

```
*****
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 <rpsvc/mount.h>
56 #include <rpsvc/nsm_addr.h>
57 #include <rpsvc/rquota.h>
```

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

2

```
58 #include <rpsvc/sm_inter.h>
59 #include <rpsvc/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 xpVtbl_list_lock: xpVtbl_list */

84 SVCXPRT_LIST *_svc_xpVtbl_list = NULL;
85 extern mutex_t xpVtbl_list_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 rpsvc_t,
92     const struct netconfig *, const struct t_bind *);

94 boolean_t
95 is_multilevel(rpcprog_t prognum)
96 {
97     /* This is a list of identified multilevel service provider */
98     if ((prognum == MOUNTPROG) || (prognum == NFS_PROGRAM) ||
99         (prognum == NFS_ACL_PROGRAM) || (prognum == NLM_PROG) ||
100        (prognum == NSM_ADDR_PROGRAM) || (prognum == RQUOTAPROG) ||
101        (prognum == 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 rpsvcbind.
184  * with the rpsvcbind. It calls svc_tli_create();
185  */
186 SVCXPRT *
187 svc_tp_create(void (*dispatch)(), const rpcprog_t prognum,
188     const rpsvc_t versnum, const struct netconfig *nconf)
189 {
190     return (svc_tp_create_bind(dispatch, prognum, versnum, nconf, NULL));
191 }

192 /*
193  * svc_tp_create_addr()
194  * Variant of svc_tp_create() that allows specifying just the
```

```

195 * the binding address, for convenience.
196 */
197 SVCXPRT *
198 svc_tp_create_addr(void (*dispatch)(), const rpcprog_t prognum,
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_LSTNBKLOG_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, prognum, versnum,
220 nconf, bindp));
221 }
222
223 static SVCXPRT *
224 svc_tp_create_bind(void (*dispatch)(), const rpcprog_t prognum,
225 const rpcvers_t versnum, const struct netconfig *nconf,
226 const struct t_bind *bindaddr)
227 {
228     SVCXPRT *xpirt;
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", prognum, versnum);
234         return (NULL);
235     }
236
237     /* Some programs need to allocate MLP for multilevel services */
238     if (is_system_labeled() && is_multilevel(prognum))
239         anon_mlp = B_TRUE;
240     xpirt = svc_tli_create_common(RPC_ANYFD, nconf, bindaddr, 0, 0,
241 anon_mlp);
242     xpirt = svc_tli_create_common(RPC_ANYFD, nconf, NULL, 0, 0, anon_mlp);
243     if (xpirt == NULL)
244         return (NULL);
245
246     (void) rpcb_unset(prognum, versnum, (struct netconfig *)nconf);
247     if (svc_reg(xpirt, prognum, versnum, dispatch, nconf) == FALSE) {
248         (void) syslog(LOG_ERR,
249             "svc_tp_create: Could not register prog %d vers %d on %s",
250 prognum, versnum, nconf->nc_netid);
251         SVC_DESTROY(xpirt);
252         return (NULL);
253     }
254     return (xpirt);
255 }
256
257 unchanged_portion_omitted
258
259 /*
260 * If fd is RPC_ANYFD, then it opens a fd for the given transport
261 * provider (nconf cannot be NULL then). If the t_state is T_UNBND and

```

```

262 * bindaddr is NON-NULL, it performs a t_bind using the bindaddr. For
263 * NULL bindaddr and Connection oriented transports, the value of qlen
264 * is set arbitrarily.
265 *
266 * If sendsz or recvsz are zero, their default values are chosen.
267 */
268 SVCXPRT *
269 svc_tli_create_common(const int ofd, const struct netconfig *nconf,
270 const struct t_bind *bindaddr, const uint_t sendsz,
271 const uint_t recvsz, boolean_t mlp_flag)
272 {
273     SVCXPRT *xpirt = NULL; /* service handle */
274     struct t_info tinfo; /* transport info */
275     struct t_bind *tres = NULL; /* bind info */
276     bool_t madefd = FALSE; /* whether fd opened here */
277     int state; /* state of the transport provider */
278     int fd = ofd;
279
280     if (fd == RPC_ANYFD) {
281         if (nconf == NULL) {
282             (void) syslog(LOG_ERR,
283                 "svc_tli_create: invalid netconfig");
284             return (NULL);
285         }
286         fd = t_open(nconf->nc_device, O_RDWR, &tinfo);
287         if (fd == -1) {
288             char errorstr[100];
289
290             __tli_sys_strerror(errorstr, sizeof (errorstr),
291 t_errno, errno);
292             (void) syslog(LOG_ERR, "svc_tli_create: could not open "
293                 "connection for %s: %s", nconf->nc_netid, errorstr);
294             return (NULL);
295         }
296         madefd = TRUE;
297         state = T_UNBND;
298     } else {
299         /*
300         * It is an open descriptor. Sync it & get the transport info.
301         */
302         if ((state = t_sync(fd)) == -1) {
303             char errorstr[100];
304
305             __tli_sys_strerror(errorstr, sizeof (errorstr),
306 t_errno, errno);
307             (void) syslog(LOG_ERR,
308                 "svc_tli_create: could not do t_sync: %s",
309 errorstr);
310             return (NULL);
311         }
312         if (t_getinfo(fd, &tinfo) == -1) {
313             char errorstr[100];
314
315             __tli_sys_strerror(errorstr, sizeof (errorstr),
316 t_errno, errno);
317             (void) syslog(LOG_ERR, "svc_tli_create: could not get "
318                 "transport information: %s", errorstr);
319             return (NULL);
320         }
321         /* Enable options of returning the ip's for udp */
322         if (nconf) {
323             int ret = 0;
324             if (strcmp(nconf->nc_netid, "udp6") == 0) {
325                 ret = __rpc_tli_set_options(fd, IPPROTO_IPV6,
326 IPV6_RECVPCKTINFO, 1);
327                 if (ret < 0) {

```

```

332     char errorstr[100];
333
334     __tli_sys_strerror(errorstr,
335     sizeof (errorstr), t_errno, errno);
336     (void) syslog(LOG_ERR,
337     "svc_tli_create: "
338     "IPV6_RECVPKTINFO(1): %s",
339     errorstr);
340     return (NULL);
341 }
342 } else if (strcmp(nconf->nc_netid, "udp") == 0) {
343     ret = __rpc_tli_set_options(fd, IPPROTO_IP,
344     IP_RECVSTADDR, 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_RECVSTADDR(1): %s", errorstr);
353         return (NULL);
354     }
355 }
356 }
357 }
358
359 /*
360  * If the fd is unbound, try to bind it.
361  * In any case, try to get its bound info in tres
362  */
363 /* LINTED pointer alignment */
364 tres = (struct t_bind *)t_alloc(fd, T_BIND, T_ADDR);
365 if (tres == NULL) {
366     (void) syslog(LOG_ERR, "svc_tli_create: No memory!");
367     goto freedata;
368 }
369
370 switch (state) {
371     bool_t tcp, exclbind;
372 case T_UNBND:
373     /* If this is a labeled system, then ask for an MLP */
374     if (is_system_labeled() &&
375     (strcmp(nconf->nc_protofmly, NC_INET) == 0 ||
376     strcmp(nconf->nc_protofmly, NC_INET6) == 0)) {
377         (void) __rpc_tli_set_options(fd, SOL_SOCKET,
378         SO_RECVUCRED, 1);
379         if (mlp_flag)
380             (void) __rpc_tli_set_options(fd, SOL_SOCKET,
381             SO_ANON_MLP, 1);
382     }
383
384     if (bindaddr) {
385         /*
386          * Services that specify a bind address typically
387          * use a fixed service (IP port) so we need to set
388          * SO_REUSEADDR to prevent bind errors on restart.
389          * SO_EXCLBIND has the following properties
390          * - an fd bound to port P via IPv4 will prevent an IPv6
391          *   bind to port P (and vice versa)
392          * - an fd bound to a wildcard IP address for port P will
393          *   prevent a more specific IP address bind to port P
394          * (see {tcp,udp}.c for details)
395          *
396          * We use the latter property to prevent hijacking of RPC
397          * services that reside at non-privileged ports.

```

```

388     */
389     if (bindaddr->addr.len != 0)
390         (void) __rpc_tli_set_options(fd, SOL_SOCKET,
391         SO_REUSEADDR, 1);
392     tcp = nconf ? (strcmp(nconf->nc_proto, NC_TCP) == 0) : 0;
393     if (nconf &&
394     (tcp || (strcmp(nconf->nc_proto, NC_UDP) == 0)) &&
395     rpc_control(__RPC_SVC_EXCLBIND_GET, &exclbind)) {
396         if (exclbind) {
397             if (__rpc_tli_set_options(fd, SOL_SOCKET,
398             SO_EXCLBIND, 1) < 0) {
399                 syslog(LOG_ERR,
400                 "svc_tli_create: can't set EXCLBIND [netid='%s']",
401                 nconf->nc_netid);
402                 goto freedata;
403             }
404         }
405     }
406     if (bindaddr) {
407         if (t_bind(fd, (struct t_bind *)bindaddr, tres) == -1) {
408             char errorstr[100];
409
410             __tli_sys_strerror(errorstr, sizeof (errorstr),
411             t_errno, errno);
412             (void) syslog(LOG_ERR,
413             "svc_tli_create: could not bind: %s",
414             errorstr);
415             goto freedata;
416         }
417         /*
418          * Should compare the addresses only if addr.len
419          * was non-zero
420          */
421         if (bindaddr->addr.len &&
422         (memcmp(bindaddr->addr.buf, tres->addr.buf,
423         (int)tres->addr.len) != 0)) {
424             (void) syslog(LOG_ERR, "svc_tli_create: could "
425             "not bind to requested address: address "
426             "mismatch");
427             goto freedata;
428         }
429     } else {
430         if (rpc_control(__RPC_SVC_LSTNBKLOG_GET, &tres->qlen)
431         == FALSE) {
432             syslog(LOG_ERR,
433             "svc_tli_create: can't get listen backlog");
434             goto freedata;
435         }
436         tres->addr.len = 0;
437         if (t_bind(fd, tres, tres) == -1) {
438             char errorstr[100];
439
440             __tli_sys_strerror(errorstr, sizeof (errorstr),
441             t_errno, errno);
442             (void) syslog(LOG_ERR,
443             "svc_tli_create: could not bind: %s",
444             errorstr);
445             goto freedata;
446         }
447     }
448 }
449
450 /*
451  * If requested, set SO_EXCLBIND on each binding.
452  *
453  * SO_EXCLBIND has the following properties
454  * - 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 }

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 }

```

```

505     }
506     }
507     break;

509 case T_IDLE:
510     if (bindaddr) {
511         /* Copy the entire stuff in tres */
512         if (tres->addr.maxlen < bindaddr->addr.len) {
513             (void) syslog(LOG_ERR,
514                 "svc_tli_create: illegal netbuf length");
515             goto freedata;
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     break;
524 case T_INREL:
525     (void) t_rcvrel(fd);
526     (void) t_sndrel(fd);
527     (void) syslog(LOG_ERR, "svc_tli_create: other side wants to "
528         "release connection");
529     goto freedata;

531 case T_INCON:
532     /* Do nothing here. Assume this is handled in rendezvous */
533     break;
534 case T_DATAXFER:
535     /*
536      * This takes care of the case where a fd
537      * is passed on which a connection has already
538      * been accepted.
539      */
540     if (t_getname(fd, &(tres->addr), LOCALNAME) == -1)
541         tres->addr.len = 0;
542     break;
543 default:
544     (void) syslog(LOG_ERR,
545         "svc_tli_create: connection in a wierd state (%d)", state);
546     goto freedata;
547 }

549 /*
550 * call transport specific function.
551 */
552 switch (tinfo.servtype) {
553 case T_COTS_ORD:
554 case T_COTS:
555     if (state == T_DATAXFER)
556         xprt = svc_fd_create_private(fd, sendsz,
557             recvsz);
558     else
559         xprt = svc_vc_create_private(fd, sendsz,
560             recvsz);
561     if (!nconf || !xprt)
562         break;
563     if ((tinfo.servtype == T_COTS_ORD) &&
564         (state != T_DATAXFER) &&
565         (strcmp(nconf->nc_protofmly, "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;

583     /* fill in the other xprt information */

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;

593     xprt->xp_rtaddr.len = 0;
594     xprt->xp_rtaddr.maxlen = __rpc_get_a_size(tinfo.addr);

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     }

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     }

621     /*
622     *   if (madeafd && (tinfo.servtype == T_CLTS))
623     *       (void) ioctl(fd, I_POP, NULL);
624     */
625     xprt_register(xprt);
626     return (xprt);

628 freedata:
629     if (madeafd)
630         (void) t_close(fd);
631     if (tres)
632         (void) t_free((char *)tres, T_BIND);
633     if (xprt) {
634         if (!madeafd) /* so that svc_destroy doesnt 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

1

```
*****
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 */
21
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 */
27
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 <rpcsvc/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>
```

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

2

```
55 #include <netconfig.h>
56 #include "smfcfg.h"
57
58 /* should really be in some global place */
59 #define DEF_WIN 30000
60 #define OPT_CHUNK 1024
61
62 int debug = 0;
63
64 #define NFS_SERVER_SVC "svc:/network/nfs/server:default"
65 #define NFS_CLIENT_SVC (char *)"svc:/network/nfs/client:default"
66
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();
82
83 /*
84  * ops vector that provides the protocol specific info and operations
85  * for share management.
86  */
87
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
118 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,
```

```

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_GE},
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     {"nfsd_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     {"nfsd_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     {"nfsd_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     {"nfsd_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     {"mountd_max_threads",
2564      "mountd_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     {"mountd_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              getipsecprotobyname.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              xdr.3nsl            \|
95              xdr_admin.3nsl      \|
96              xdr_complex.3nsl    \|
97              xdr_create.3nsl     \|
98              xdr_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             authnone_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      endrpercent.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      getipsecalgbynum.3nsl  //
153      getipsecprotobynum.3nsl //
154      getnetconfigent.3nsl   //
155      getnetname.3nsl        //
156      getrpcbyname_r.3nsl    //
157      getrpcbynumber.3nsl    //
158      getrpcbynumber_r.3nsl  //
159      getrpercent.3nsl       //
160      getrpercent_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_sperror.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      setrpercent.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      svcerr_auth.3nsl      //
245      svcerr_decode.3nsl    //
246      svcerr_noproc.3nsl    //
247      svcerr_noprogram.3nsl //
248      svcerr_progvers.3nsl  //
249      svcerr_systemerr.3nsl //
250      svcerr_weakauth.3nsl  //
251      svcfld_create.3nsl    //
252      svcraw_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_replymsg.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 xdrmem_create.3nsl //
303 xdrrec_create.3nsl //
304 xdrrec_endofrecord.3nsl //
305 xdrrec_eof.3nsl //
306 xdrrec_readbytes.3nsl //
307 xdrrec_skiprecord.3nsl //
308 xdrstdio_create.3nsl //
309 xpvt_register.3nsl //
310 xpvt_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 //

```

```

323 undial.3nsl := LINKSRC = dial.3nsl

325 endhostent.3nsl := LINKSRC = gethostbyname.3nsl
326 gethostbyaddr.3nsl := LINKSRC = gethostbyname.3nsl
327 gethostbyaddr_r.3nsl := LINKSRC = gethostbyname.3nsl
328 gethostbyname_r.3nsl := LINKSRC = gethostbyname.3nsl
329 gethostent.3nsl := LINKSRC = gethostbyname.3nsl
330 gethostent_r.3nsl := LINKSRC = gethostbyname.3nsl
331 sethostent.3nsl := LINKSRC = gethostbyname.3nsl

333 freeipsecalgent.3nsl := LINKSRC = getipsecalgbyname.3nsl
334 getipsecalgbynum.3nsl := LINKSRC = getipsecalgbyname.3nsl

336 getipsecprotobynum.3nsl := LINKSRC = getipsecprotobynum.3nsl

338 endnetconfig.3nsl := LINKSRC = getnetconfig.3nsl
339 freenetconfigent.3nsl := LINKSRC = getnetconfig.3nsl
340 getnetconfigent.3nsl := LINKSRC = getnetconfig.3nsl
341 nc_perror.3nsl := LINKSRC = getnetconfig.3nsl
342 nc_sperror.3nsl := LINKSRC = getnetconfig.3nsl
343 setnetconfig.3nsl := LINKSRC = getnetconfig.3nsl

345 endnetpath.3nsl := LINKSRC = getnetpath.3nsl
346 setnetpath.3nsl := LINKSRC = getnetpath.3nsl

348 getsecretkey.3nsl := LINKSRC = getpublickey.3nsl
349 publickey.3nsl := LINKSRC = getpublickey.3nsl

351 endrpcnt.3nsl := LINKSRC = getrpcbyname.3nsl
352 getrpcbyname_r.3nsl := LINKSRC = getrpcbyname.3nsl
353 getrpcbynumber.3nsl := LINKSRC = getrpcbyname.3nsl
354 getrpcbynumber_r.3nsl := LINKSRC = getrpcbyname.3nsl
355 getrpcnt.3nsl := LINKSRC = getrpcbyname.3nsl
356 getrpcnt_r.3nsl := LINKSRC = getrpcbyname.3nsl
357 setrpcnt.3nsl := LINKSRC = getrpcbyname.3nsl

359 netdir_free.3nsl := LINKSRC = netdir.3nsl
360 netdir_getbyaddr.3nsl := LINKSRC = netdir.3nsl
361 netdir_getbyname.3nsl := LINKSRC = netdir.3nsl
362 netdir_mergeaddr.3nsl := LINKSRC = netdir.3nsl
363 netdir_options.3nsl := LINKSRC = netdir.3nsl
364 netdir_perror.3nsl := LINKSRC = netdir.3nsl
365 netdir_sperror.3nsl := LINKSRC = netdir.3nsl
366 taddr2uaddr.3nsl := LINKSRC = netdir.3nsl
367 uaddr2taddr.3nsl := LINKSRC = netdir.3nsl

369 auth_destroy.3nsl := LINKSRC = rpc_clnt_auth.3nsl
370 authnone_create.3nsl := LINKSRC = rpc_clnt_auth.3nsl
371 authsys_create.3nsl := LINKSRC = rpc_clnt_auth.3nsl
372 authsys_create_default.3nsl := LINKSRC = rpc_clnt_auth.3nsl

374 clnt_call.3nsl := LINKSRC = rpc_clnt_calls.3nsl
375 clnt_freeres.3nsl := LINKSRC = rpc_clnt_calls.3nsl
376 clnt_geterr.3nsl := LINKSRC = rpc_clnt_calls.3nsl
377 clnt_perrno.3nsl := LINKSRC = rpc_clnt_calls.3nsl
378 clnt_perror.3nsl := LINKSRC = rpc_clnt_calls.3nsl
379 clnt_send.3nsl := LINKSRC = rpc_clnt_calls.3nsl
380 clnt_sperrno.3nsl := LINKSRC = rpc_clnt_calls.3nsl
381 clnt_sperror.3nsl := LINKSRC = rpc_clnt_calls.3nsl
382 rpc_broadcast.3nsl := LINKSRC = rpc_clnt_calls.3nsl
383 rpc_broadcast_exp.3nsl := LINKSRC = rpc_clnt_calls.3nsl
384 rpc_call.3nsl := LINKSRC = rpc_clnt_calls.3nsl

386 clnt_control.3nsl := LINKSRC = rpc_clnt_create.3nsl
387 clnt_create.3nsl := LINKSRC = rpc_clnt_create.3nsl

```

```

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

403 rpc_gss_get_mech_info.3nsl := LINKSRC = rpc_gss_get_mechanisms.3nsl
404 rpc_gss_get_versions.3nsl := LINKSRC = rpc_gss_get_mechanisms.3nsl
405 rpc_gss_is_installed.3nsl := LINKSRC = rpc_gss_get_mechanisms.3nsl

407 rpc_gss_svc_max_data_length.3nsl := LINKSRC = rpc_gss_max_data_length.3nsl

409 rpc_gss_qop_to_num.3nsl      := LINKSRC = rpc_gss_mech_to_oid.3nsl

411 authdes_create.3nsl         := LINKSRC = rpc_soc.3nsl
412 authunix_create.3nsl        := LINKSRC = rpc_soc.3nsl
413 authunix_create_default.3nsl := LINKSRC = rpc_soc.3nsl
414 callrpc.3nsl                := LINKSRC = rpc_soc.3nsl
415 clnt_broadcast.3nsl         := LINKSRC = rpc_soc.3nsl
416 clntraw_create.3nsl         := LINKSRC = rpc_soc.3nsl
417 clnttcp_create.3nsl         := LINKSRC = rpc_soc.3nsl
418 clntudp_bufcreate.3nsl      := LINKSRC = rpc_soc.3nsl
419 clntudp_create.3nsl         := LINKSRC = rpc_soc.3nsl
420 get_myaddress.3nsl          := LINKSRC = rpc_soc.3nsl
421 getrpcport.3nsl             := LINKSRC = rpc_soc.3nsl
422 pmap_getmaps.3nsl           := LINKSRC = rpc_soc.3nsl
423 pmap_getport.3nsl           := LINKSRC = rpc_soc.3nsl
424 pmap_rmtcall.3nsl           := LINKSRC = rpc_soc.3nsl
425 pmap_set.3nsl                := LINKSRC = rpc_soc.3nsl
426 pmap_unset.3nsl             := LINKSRC = rpc_soc.3nsl
427 registerrpc.3nsl           := LINKSRC = rpc_soc.3nsl
428 svc_fds.3nsl                 := LINKSRC = rpc_soc.3nsl
429 svc_getcaller.3nsl           := LINKSRC = rpc_soc.3nsl
430 svc_getreq.3nsl              := LINKSRC = rpc_soc.3nsl
431 svc_register.3nsl            := LINKSRC = rpc_soc.3nsl
432 svc_unregister.3nsl          := LINKSRC = rpc_soc.3nsl
433 svcd_create.3nsl             := LINKSRC = rpc_soc.3nsl
434 svcraw_create.3nsl           := LINKSRC = rpc_soc.3nsl
435 svctcp_create.3nsl           := LINKSRC = rpc_soc.3nsl
436 svcdudp_bufcreate.3nsl      := LINKSRC = rpc_soc.3nsl
437 svcdudp_create.3nsl         := LINKSRC = rpc_soc.3nsl
438 xdr_authunix_parms.3nsl      := LINKSRC = rpc_soc.3nsl

440 svc_dg_enablecache.3nsl     := LINKSRC = rpc_svc_calls.3nsl
441 svc_done.3nsl                := LINKSRC = rpc_svc_calls.3nsl
442 svc_exit.3nsl                := LINKSRC = rpc_svc_calls.3nsl
443 svc_fd_negotiate_ucred.3nsl := LINKSRC = rpc_svc_calls.3nsl
444 svc_fdset.3nsl               := LINKSRC = rpc_svc_calls.3nsl
445 svc_freeargs.3nsl            := LINKSRC = rpc_svc_calls.3nsl
446 svc_getargs.3nsl             := LINKSRC = rpc_svc_calls.3nsl
447 svc_getcallerucred.3nsl     := LINKSRC = rpc_svc_calls.3nsl
448 svc_getreq_common.3nsl      := LINKSRC = rpc_svc_calls.3nsl
449 svc_getreq_poll.3nsl        := LINKSRC = rpc_svc_calls.3nsl
450 svc_getreqset.3nsl           := LINKSRC = rpc_svc_calls.3nsl
451 svc_getrpccaller.3nsl       := LINKSRC = rpc_svc_calls.3nsl
452 svc_max_pollfd.3nsl         := LINKSRC = rpc_svc_calls.3nsl
453 svc_pollfd.3nsl              := LINKSRC = rpc_svc_calls.3nsl

```

```

454 svc_run.3nsl                := LINKSRC = rpc_svc_calls.3nsl
455 svc_sendreply.3nsl          := LINKSRC = rpc_svc_calls.3nsl

457 svc_control.3nsl           := LINKSRC = rpc_svc_create.3nsl
458 svc_create.3nsl             := LINKSRC = rpc_svc_create.3nsl
459 svc_destroy.3nsl            := LINKSRC = rpc_svc_create.3nsl
460 svc_dg_create.3nsl          := LINKSRC = rpc_svc_create.3nsl
461 svc_door_create.3nsl        := LINKSRC = rpc_svc_create.3nsl
462 svc_fd_create.3nsl          := LINKSRC = rpc_svc_create.3nsl
463 svc_raw_create.3nsl         := LINKSRC = rpc_svc_create.3nsl
464 svc_tli_create.3nsl         := LINKSRC = rpc_svc_create.3nsl
465 svc_tp_create.3nsl          := LINKSRC = rpc_svc_create.3nsl
466 svc_tp_create_addr.3nsl    := LINKSRC = rpc_svc_create.3nsl
467 svc_vc_create.3nsl          := LINKSRC = rpc_svc_create.3nsl

469 svcerr_auth.3nsl            := LINKSRC = rpc_svc_err.3nsl
470 svcerr_decode.3nsl          := LINKSRC = rpc_svc_err.3nsl
471 svcerr_noproc.3nsl          := LINKSRC = rpc_svc_err.3nsl
472 svcerr_noprog.3nsl         := LINKSRC = rpc_svc_err.3nsl
473 svcerr_progvers.3nsl        := LINKSRC = rpc_svc_err.3nsl
474 svcerr_systemerr.3nsl      := LINKSRC = rpc_svc_err.3nsl
475 svcerr_weakauth.3nsl       := LINKSRC = rpc_svc_err.3nsl

477 svc_add_input.3nsl          := LINKSRC = rpc_svc_input.3nsl
478 svc_remove_input.3nsl       := LINKSRC = rpc_svc_input.3nsl

480 rpc_reg.3nsl                 := LINKSRC = rpc_svc_reg.3nsl
481 svc_auth_reg.3nsl           := LINKSRC = rpc_svc_reg.3nsl
482 svc_reg.3nsl                 := LINKSRC = rpc_svc_reg.3nsl
483 svc_unreg.3nsl               := LINKSRC = rpc_svc_reg.3nsl
484 xpirt_register.3nsl          := LINKSRC = rpc_svc_err.3nsl
485 xpirt_unregister.3nsl       := LINKSRC = rpc_svc_reg.3nsl

487 xdr_accepted_reply.3nsl     := LINKSRC = rpc_xdr.3nsl
488 xdr_authsys_parms.3nsl     := LINKSRC = rpc_xdr.3nsl
489 xdr_callhdr.3nsl            := LINKSRC = rpc_xdr.3nsl
490 xdr_callmsg.3nsl            := LINKSRC = rpc_xdr.3nsl
491 xdr_opaque_auth.3nsl        := LINKSRC = rpc_xdr.3nsl
492 xdr_rejected_reply.3nsl     := LINKSRC = rpc_xdr.3nsl
493 xdr_replymsg.3nsl           := LINKSRC = rpc_xdr.3nsl

495 rpcb_getaddr.3nsl           := LINKSRC = rpcbind.3nsl
496 rpcb_getmaps.3nsl           := LINKSRC = rpcbind.3nsl
497 rpcb_gettime.3nsl           := LINKSRC = rpcbind.3nsl
498 rpcb_rmtcall.3nsl           := LINKSRC = rpcbind.3nsl
499 rpcb_set.3nsl                := LINKSRC = rpcbind.3nsl
500 rpcb_unset.3nsl              := LINKSRC = rpcbind.3nsl

502 authdes_getucred.3nsl       := LINKSRC = secure_rpc.3nsl
503 authdes_seccreate.3nsl      := LINKSRC = secure_rpc.3nsl
504 getnetname.3nsl              := LINKSRC = secure_rpc.3nsl
505 host2netname.3nsl           := LINKSRC = secure_rpc.3nsl
506 key_decryptsession.3nsl     := LINKSRC = secure_rpc.3nsl
507 key_encryptsession.3nsl     := LINKSRC = secure_rpc.3nsl
508 key_gendes.3nsl              := LINKSRC = secure_rpc.3nsl
509 key_secretkey_is_set.3nsl   := LINKSRC = secure_rpc.3nsl
510 key_setsecret.3nsl           := LINKSRC = secure_rpc.3nsl
511 netname2host.3nsl            := LINKSRC = secure_rpc.3nsl
512 netname2user.3nsl           := LINKSRC = secure_rpc.3nsl
513 user2netname.3nsl           := LINKSRC = secure_rpc.3nsl

515 xdr_control.3nsl             := LINKSRC = xdr_admin.3nsl
516 xdr_getpos.3nsl             := LINKSRC = xdr_admin.3nsl
517 xdr_inline.3nsl             := LINKSRC = xdr_admin.3nsl
518 xdr_setpos.3nsl              := LINKSRC = xdr_admin.3nsl
519 xdr_sizeof.3nsl              := LINKSRC = xdr_admin.3nsl

```



```
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)
```

```

*****
14979 Mon Jul 17 16:15:19 2017
new/usr/src/man/man3nsl/rpc_svc_create.3nsl
8330 Add svc_tp_create_addr to libnsl
Reviewed by: Paul Dagnelie <pcdd@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.
3  .\" Copyright 2014 Nexenta Systems, Inc. All Rights Reserved.
4  .\" Copyright 1989 AT&T
5  .\" Copyright (C) 2005, Sun Microsystems, Inc. All Rights Reserved.
6  .\" The contents of this file are subject to the terms of the Common Development
7  .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
8  .\" When distributing Covered Code, include this CDDL HEADER in each file and in
9  .TH RPC_SVC_CREATE 3NSL "Jun 19, 2017"
10 .TH RPC_SVC_CREATE 3NSL "May 18, 2017"
11 .SH NAME
12 rpc_svc_create, svc_control, svc_create, svc_destroy, svc_dg_create,
13 svc_fd_create, svc_raw_create, svc_tli_create, svc_tp_create,
14 svc_tp_create_addr, svc_vc_create,
15 svc_fd_create, svc_raw_create, svc_tli_create, svc_tp_create, svc_vc_create,
16 svc_door_create \- server handle creation routines
17 .SH SYNOPSIS
18 .LP
19 #include <rpc/rpc.h>
20 .fi
21
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 \fIprognum\fR, \fBconst rpcvers_t\f
26     \fBconst char *\fR\fIinetype\fR);
27 .fi
28
29 .LP
30 .nf
31 \fBvoid\fR \fBsvc_destroy\fR(\fBconst struct svc_r *)\fR;
32 .fi
33
34 .LP
35 .nf
36 \fBvoid\fR \fBsvc_dg_create\fR(\fBconst int\fR \fIifildes\fR, \fBconst uint_t
37     \fBconst uint_t\fR \fIirecvsz\fR);
38 .fi
39
40 .LP
41 .nf
42 \fBvoid\fR \fBsvc_fd_create\fR(\fBconst int\fR \fIifildes\fR, \fBconst uint_t
43     \fBconst uint_t\fR \fIirecvsz\fR);
44 .fi
45
46 .LP
47 .nf
48 \fBvoid\fR \fBsvc_raw_create\fR(void);
49 .fi
50
51 .LP
52 .nf
53 \fBvoid\fR \fBsvc_tli_create\fR(\fBconst int\fR \fIifildes\fR, \fBconst struc
54     \fBconst struct t_bind *\fR\fIbind_info\fR, \fBconst uint_t\fR \fIisendsz\fR
55     \fBconst struct t_bind *\fR\fIbind_addr\fR, \fBconst uint_t\fR \fIisendsz\fR

```

```

56 .fi
57
58 .LP
59 .nf
60 \fBvoid\fR \fBsvc_tp_create\fR(\fBconst void (*\fR\fIdispatch\fR)
61     (const struct svc_req *, const SVCXPRT *), \fBconst rpcprog_t\fR \fIprognum
62     \fBconst rpcvers_t\fR \fIversnum\fR, \fBconst struct netconfig *\fR\fIinetco
63 .fi
64
65 .LP
66 .nf
67 \fBvoid\fR \fBsvc_tp_create_addr\fR(\fBconst void (*\fR\fIdispatch\fR)
68     (const struct svc_req *, const SVCXPRT *), \fBconst rpcprog_t\fR \fIprognum
69     \fBconst rpcvers_t\fR \fIversnum\fR, \fBconst struct netconfig *\fR\fIinetco
70     \fBconst struct netbuf *\fR\fIbind_addr\fR)
71 );
72 .fi
73
74 .LP
75 .nf
76 \fBvoid\fR \fBsvc_vc_create\fR(\fBconst int\fR \fIifildes\fR, \fBconst uint_t
77     \fBconst uint_t\fR \fIirecvsz\fR);
78 .fi
79
80 .LP
81 .nf
82 \fBvoid\fR \fBsvc_door_create\fR(\fBvoid (*\fR\fIdispatch\fR)(struct svc_req
83     \fBconst rpcprog_t\fR \fIprognum\fR, \fBconst rpcvers_t\fR \fIversnum\fR,
84     \fBconst uint_t\fR \fIisendsz\fR);
85 .fi
86
87 .SH DESCRIPTION
88 .LP
89 These routines are part of the \fBbrpc\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 \fBbrpc\fR(3NSL) for the definition of the \fBsvc_t\fR data structure.
96 .sp
97 .ne 2
98 .na
99 \fBsvc_control()\fR
100 .ad
101 .RS 15n
102 .RS 21n
103 A function to change or retrieve information about a service object. \fIreq\fR
104 indicates the type of operation and \fIinfo\fR is a pointer to the information.
105 The supported values of \fIreq\fR, their argument types, and what they do are:
106 .sp
107 .na
108 \fBsvc_get_versquiet\fR
109 .ad
110 .RS 10n
111 .RS 25n
112 If a request is received for a program number served by this server but the
113 version number is outside the range registered with the server, an
114 \fBPROGVERS_MISMATCH\fR error will normally be returned. \fIinfo\fR should
115 be a pointer to an integer. Upon successful completion of the
116 \fBsvc_get_versquiet\fR request, \fIinfo\fR contains an integer which
117 describes the server's current behavior: \fB0\fR indicates normal server
118 behavior, that is, an \fBPROGVERS_MISMATCH\fR error will be returned.
119 \fB1\fR indicates that the out of range request will be silently ignored.

```

```

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 \fB\FRPC_PROGVERSMISMATCH\fR error will normally be returned. It is sometimes
130 desirable to change this behavior. \fI*info\fR should be a pointer to an integer
131 which is either \fB0\fR, indicating normal server behavior and an
132 \fB\FRPC_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 \fB\FRPC\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 \fI*info\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 Upon successful completion of the \fBSVCGET_RECVERRHANDLER\fR request,
192 \fI*info\fR contains the address of the handler for receiving errors. Upon
193 failure, \fI*info\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
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
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 \fBbrpcbind\fR service (see
240 \fBbrpcbind\fR(1M)). \fBdispatch\fR is called when there is a remote procedure
241 call for the given \fBprognum\fR and \fBversnum\fR; this requires calling
242 \fBsvc_run()\fR (see \fBsvc_run()\fR in \fBbrpc_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
242 .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
253 .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 \fBrcbind\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
271 .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
285 .RS 21n
296 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 \fBclnt_raw_create()\fR in \fBrc_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
303 .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
308 \fIbindaddr\fR is non-null \fIfildes\fR is bound to the address specified by
309 \fIbindaddr\fR, otherwise \fIfildes\fR is bound to a default address chosen by
320 the transport. In the case where the default address is chosen, the number of
321 outstanding connect requests is set to 8 for connection-oriented transports.
322 The user may specify the size of the send and receive buffers with the
323 parameters \fIsendsz\fR and \fIrecvsz\fR \fIi\fR values of \fB0\fR choose
324 suitable defaults. This routine returns \fINULL\fR if it fails, and an error
325 message is logged. The server is not registered with the \fBrcbind\fR(1M)
326 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 \fBrcbind\fR service.
337 \fIdispatch\fR is called when there is a remote procedure call for the given
338 \fIprognum\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 \fBrcbind\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 \fIprognum\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
379 This routine creates an RPC server handle over doors for the given program
380 \fIprognum\fR and version \fIversnum\fR and returns a pointer to it.
381 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 a value for \fBrpc_createerr\fR. The server is not
387 registered with \fBrpcbind\fR(1M). The \fBVCSET_CONNMAXREC\fR and
388 \fBVCGET_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)

```

```

*****
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 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  .\" 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 nfsd 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 nfsd ,
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

```

```

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 nfsd .
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

```

```

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 .Pq 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

1

```
*****
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 #
11 #
12 #
13 # Copyright 2011, Richard Lowe
14 # Copyright 2017 Nexenta Systems, Inc. All rights reserved.
15 # Copyright 2012 Nexenta Systems, Inc. All rights reserved.
16 #
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/rpcbnd.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
```

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

2

```
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_rcvudata.3nsl
76 file path=usr/share/man/man3nsl/t_rcvuderr.3nsl
77 file path=usr/share/man/man3nsl/t_rcvv.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_getucred.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/athunix_create.3nsl target=rpc_soc.3nsl
106 link path=usr/share/man/man3nsl/athunix_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
```

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

3

```

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_sperrno.3nsl target=rpc_clnt_calls.3nsl
136 link path=usr/share/man/man3nsl/clnt_sperror.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/ndhostent.3nsl target=gethostbyname.3nsl
150 link path=usr/share/man/man3nsl/ndnetconfig.3nsl target=getnetconfig.3nsl
151 link path=usr/share/man/man3nsl/ndnetpath.3nsl target=getnetpath.3nsl
152 link path=usr/share/man/man3nsl/ndrpcnt.3nsl target=getrpcbyname.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/getipsecalgbyname.3nsl \
163   target=getipsecalgbyname.3nsl
164 link path=usr/share/man/man3nsl/getipsecprotobyname.3nsl \
165   target=getipsecprotobyname.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/getrpcbyname_r.3nsl target=getrpcbyname.3nsl
169 link path=usr/share/man/man3nsl/getrpcbyname_r.3nsl target=getrpcbyname.3nsl
170 link path=usr/share/man/man3nsl/getrpcbyname_r_r.3nsl target=getrpcbyname.3nsl
171 link path=usr/share/man/man3nsl/getrpccent.3nsl target=getrpcbyname.3nsl
172 link path=usr/share/man/man3nsl/getrpccent_r.3nsl target=getrpcbyname.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_secretkey_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_sperror.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

```

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

4

```

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/registererrp.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/setrpccent.3nsl target=getrpcbyname.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_progvers.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/undial.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_replymsg.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=yplnt.3nsl
339 link path=usr/share/man/man3nsl/yp_bind.3nsl target=yplnt.3nsl
340 link path=usr/share/man/man3nsl/yp_first.3nsl target=yplnt.3nsl
341 link path=usr/share/man/man3nsl/yp_get_default_domain.3nsl target=yplnt.3nsl
342 link path=usr/share/man/man3nsl/yp_master.3nsl target=yplnt.3nsl
343 link path=usr/share/man/man3nsl/yp_match.3nsl target=yplnt.3nsl
344 link path=usr/share/man/man3nsl/yp_next.3nsl target=yplnt.3nsl
345 link path=usr/share/man/man3nsl/yp_order.3nsl target=yplnt.3nsl
346 link path=usr/share/man/man3nsl/yp_unbind.3nsl target=yplnt.3nsl
347 link path=usr/share/man/man3nsl/yperr_string.3nsl target=yplnt.3nsl
348 link path=usr/share/man/man3nsl/ypprot_err.3nsl target=yplnt.3nsl

```

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

1

```
*****
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-existent) 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  */
```

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

2

```
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)(); -- dispatch routine
907  * const rpcprog_t prognum; -- program number
908  * const rpcvers_t versnum; -- version number
909  * const char *nettype; -- network type
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)(); -- dispatch routine
921  * const rpcprog_t prognum; -- program number
922  * const rpcvers_t versnum; -- version number
923  * const struct netconfig *nconf; -- netconfig structure
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)(); -- dispatch routine
936  * const rpcprog_t prognum; -- program number
937  * const rpcvers_t versnum; -- version number
938  * const struct netconfig *nconf; -- netconfig structure
939  * const struct netbuf *addr; -- address to bind
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);
```

```

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      */

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 */

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 */

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 */

984 /*
985 * Memory based rpc (for speed check and testing)
986 */
987 extern SVCXPRT *svc_raw_create(void);

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 prognum;      -- program number
998 *      const rpcvers_t versnum;     -- version number
999 *      const uint_t sendsize;        -- send buffer size
1000 */

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 */

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__ */

1028 extern boolean_t is_multilevel(rpcprog_t);

1030 #ifdef PORTMAP
1031 /* For backward compatibility */
1032 #include <rpc/svc_soc.h>
1033 #endif /* PORTMAP */

1035 /*
1036 * For user level MT hot server functions
1037 */

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 */

1046 #ifdef __STDC__
1047 extern void svc_done(SVCXPRT *);
1048 #else
1049 extern void svc_done();
1050 #endif /* __STDC__ */

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;

```

unchanged portion omitted