```
**********************************************************
   36052 Sun Mar 18 01:12:57 2018
new/usr/src/Makefile.master
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

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

  22 #
  23 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
  24 # Copyright (c) 2012 by Delphix. All rights reserved.
  25 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
  26 # Copyright 2015, OmniTI Computer Consulting, Inc. All rights reserved.
  27 # Copyright 2015 Gary Mills
  28 # Copyright 2015 Igor Kozhukhov <ikozhukhov@gmail.com>
  29 # Copyright 2016 Toomas Soome <tsoome@me.com>
  30 #

  32 #
  33 # Makefile.master, global definitions for system source
  34 #
  35 ROOT=           /proto

  37 #
  38 # Adjunct root, containing an additional proto area to be used for headers
  39 # and libraries.
  40 #
  41 ADJUNCT_PROTO=

  43 #
  44 # Adjunct for building things that run on the build machine.
  45 #
  46 NATIVE_ADJUNCT= /usr

  48 #
  49 # RELEASE_BUILD should be cleared for final release builds.
  50 # NOT_RELEASE_BUILD is exactly what the name implies.
  51 #
  52 # __GNUC toggles the building of ON components using gcc and related tools.
  53 # Normally set to '#', set it to '' to do gcc build.
  54 #
  55 # The declaration POUND_SIGN is always '#'. This is needed to get around the
  56 # make feature that '#' is always a comment delimiter, even when escaped or
  57 # quoted. We use this macro expansion method to get POUND_SIGN rather than
  58 # always breaking out a shell because the general case can cause a noticable
  59 # slowdown in build times when so many Makefiles include Makefile.master.
  60 #
  61 # While the majority of users are expected to override the setting below
```



```
  62 # with an env file (via nightly or bldenv), if you aren't building that way
  63 # (ie, you're using "ws" or some other bootstrapping method) then you need
  64 # this definition in order to avoid the subshell invocation mentioned above.
  65 #

  67 PRE_POUND=                              pre\#
  68 POUND_SIGN=                             $(PRE_POUND:pre\%=%)

  70 NOT_RELEASE_BUILD=
  71 RELEASE_BUILD=                          $(POUND_SIGN)
  72 $(RELEASE_BUILD)NOT_RELEASE_BUILD=      $(POUND_SIGN)
  73 PATCH_BUILD=                            $(POUND_SIGN)

  75 # SPARC_BLD is '#' for an Intel build.
  76 # INTEL_BLD is '#' for a Sparc build.
  77 SPARC_BLD_1=    $(MACH:i386=$(POUND_SIGN))
  78 SPARC_BLD=      $(SPARC_BLD_1:sparc=)
  79 INTEL_BLD_1=    $(MACH:sparc=$(POUND_SIGN))
  80 INTEL_BLD=      $(INTEL_BLD_1:i386=)

  82 # The variables below control the compilers used during the build.
  83 # There are a number of permutations.
  84 #
  85 # __GNUC and __SUNC control (and indicate) the primary compiler.  Whichever
  86 # one is not POUND_SIGN is the primary, with the other as the shadow.  They
  87 # may also be used to control entirely compiler-specific Makefile assignments.
  88 # __GNUC and GCC are the default.
  89 #
  90 # __GNUC64 indicates that the 64bit build should use the GNU C compiler.
  91 # There is no Sun C analogue.
  92 #
  93 # The following version-specific options are operative regardless of which
  94 # compiler is primary, and control the versions of the given compilers to be
  95 # used.  They also allow compiler-version specific Makefile fragments.
  96 #

  98 __SUNC=                 $(POUND_SIGN)
  99 $(__SUNC)__GNUC=        $(POUND_SIGN)
 100 __GNUC64=               $(__GNUC)

 102 # Allow build-time "configuration" to enable or disable some things.
 103 # The default is POUND_SIGN, meaning "not enabled". If the environment
 104 # passes in an override like ENABLE_SMB_PRINTING= (empty) that will
 105 # uncomment things in the lower Makefiles to enable the feature.
 106 ENABLE_SMB_PRINTING=    $(POUND_SIGN)

 108 # CLOSED is the root of the tree that contains source which isn't released
 109 # as open source
 110 CLOSED=         $(SRC)/../closed

 112 # BUILD_TOOLS is the root of all tools including compilers.
 113 # ONBLD_TOOLS is the root of all the tools that are part of SUNWonbld.

 115 BUILD_TOOLS=            /ws/onnv-tools
 116 ONBLD_TOOLS=            $(BUILD_TOOLS)/onbld

 118 # define runtime JAVA_HOME, primarily for cmd/pools/poold
 119 JAVA_HOME=      /usr/java
 120 # define buildtime JAVA_ROOT
 121 JAVA_ROOT=      /usr/java
 122 # define to build with JAVA 8
 123 BLD_JAVA_8=     $(POUND_SIGN)

 125 GCC_ROOT=       /opt/gcc/4.4.4
 126 GCCLIBDIR=      $(GCC_ROOT)/lib
 127 GCCLIBDIR64=    $(GCC_ROOT)/lib/$(MACH64)
```

```
129 DOCBOOK_XSL_ROOT=       /usr/share/sgml/docbook/xsl-stylesheets

131 RPCGEN=         /usr/bin/rpcgen
132 STABS=          $(ONBLD_TOOLS)/bin/$(MACH)/stabs
133 ELFEXTRACT=     $(ONBLD_TOOLS)/bin/$(MACH)/elfextract
134 MBH_PATCH=      $(ONBLD_TOOLS)/bin/$(MACH)/mbh_patch
135 BTXLD=          $(ONBLD_TOOLS)/bin/$(MACH)/btxld
136 VTFONTCVT=      $(ONBLD_TOOLS)/bin/$(MACH)/vtfontcvt
137 # echo(1) and true(1) are specified without absolute paths, so that the shell
138 # spawned by make(1) may use the built-in versions.  This is minimally
139 # problematic, as the shell spawned by make(1) is known and under control, the
140 # only risk being if the shell falls back to $PATH.
141 #
142 # We specifically want an echo(1) that does interpolation of escape sequences,
143 # which ksh93, /bin/sh, and bash will all provide.
144 ECHO=           echo
145 TRUE=           true
146 INS=            $(ONBLD_TOOLS)/bin/$(MACH)/install
147 SYMLINK=        /usr/bin/ln -s
148 LN=             /usr/bin/ln
149 MKDIR=          /usr/bin/mkdir
150 CHMOD=          /usr/bin/chmod
151 MV=             /usr/bin/mv -f
152 RM=             /usr/bin/rm -f
153 CUT=            /usr/bin/cut
154 NM=             /usr/ccs/bin/nm
155 DIFF=           /usr/bin/diff
156 GREP=           /usr/bin/grep
157 EGREP=          /usr/bin/egrep
158 ELFWRAP=        /usr/bin/elfwrap
159 KSH93=          /usr/bin/ksh93
160 SED=            /usr/bin/sed
161 AWK=            /usr/bin/nawk
162 CP=             /usr/bin/cp -f
163 MCS=            /usr/ccs/bin/mcs
164 CAT=            /usr/bin/cat
165 ELFDUMP=        /usr/ccs/bin/elfdump
166 M4=             /usr/bin/m4
167 STRIP=          /usr/ccs/bin/strip
168 LEX=            /usr/ccs/bin/lex
169 FLEX=           /usr/bin/flex
170 YACC=           /usr/ccs/bin/yacc
171 CPP=            /usr/lib/cpp
172 ANSI_CPP=       $(GCC_ROOT)/bin/cpp
173 JAVAC=          $(JAVA_ROOT)/bin/javac
174 JAVAH=          $(JAVA_ROOT)/bin/javah
175 JAVADOC=        $(JAVA_ROOT)/bin/javadoc
176 RMIC=           $(JAVA_ROOT)/bin/rmic
177 JAR=            $(JAVA_ROOT)/bin/jar
178 CTFCONVERT=     $(ONBLD_TOOLS)/bin/$(MACH)/ctfconvert
179 CTFMERGE=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfmerge
180 CTFSTABS=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfstabs
181 CTFSTRIP=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfstrip
182 NDRGEN=         $(ONBLD_TOOLS)/bin/$(MACH)/ndrgen
183 GENOFFSETS=     $(ONBLD_TOOLS)/bin/genoffsets
184 XREF=           $(ONBLD_TOOLS)/bin/xref
185 FIND=           /usr/bin/find
186 PERL=           /usr/bin/perl
187 PERL_VERSION=   5.10.0
188 PERL_PKGVERS=   -510
189 PERL_ARCH =             i86pc-solaris-64int
190 $(SPARC_BLD)PERL_ARCH = sun4-solaris-64int
191 PYTHON_VERSION= 2.7
192 PYTHON_PKGVERS= -27
193 PYTHON=         /usr/bin/python$(PYTHON_VERSION)
```



```
194 SORT=           /usr/bin/sort
195 TOUCH=          /usr/bin/touch
196 WC=             /usr/bin/wc
197 XARGS=          /usr/bin/xargs
198 ELFEDIT=        /usr/bin/elfedit
199 DTRACE=         /usr/sbin/dtrace -xnolibs
200 UNIQ=           /usr/bin/uniq
201 TAR=            /usr/bin/tar
202 ASTBINDIR=      /usr/ast/bin
203 MSGCC=          $(ASTBINDIR)/msgcc
204 MSGFMT=         /usr/bin/msgfmt -s
205 LCDEF=          $(ONBLD_TOOLS)/bin/$(MACH)/localedef
206 TIC=            $(ONBLD_TOOLS)/bin/$(MACH)/tic
207 ZIC=            $(ONBLD_TOOLS)/bin/$(MACH)/zic

209 FILEMODE=       644
210 DIRMODE=        755

212 # Declare that nothing should be built in parallel.
213 # Individual Makefiles can use the .PARALLEL target to declare otherwise.
214 .NO_PARALLEL:

216 # For stylistic checks
217 #
218 # Note that the X and C checks are not used at this time and may need
219 # modification when they are actually used.
220 #
221 CSTYLE=         $(ONBLD_TOOLS)/bin/cstyle
222 CSTYLE_TAIL=
223 HDRCHK=         $(ONBLD_TOOLS)/bin/hdrchk
224 HDRCHK_TAIL=
225 JSTYLE=         $(ONBLD_TOOLS)/bin/jstyle

227 DOT_H_CHECK=    \
228         @$(ECHO) "checking $<"; $(CSTYLE) $< $(CSTYLE_TAIL); \
229         $(HDRCHK) $< $(HDRCHK_TAIL)

231 DOT_X_CHECK=    \
232         @$(ECHO) "checking $<"; $(RPCGEN) -C -h $< | $(CSTYLE) $(CSTYLE_TAIL); \
233         $(RPCGEN) -C -h $< | $(HDRCHK) $< $(HDRCHK_TAIL)

235 DOT_C_CHECK=    \
236         @$(ECHO) "checking $<"; $(CSTYLE) $< $(CSTYLE_TAIL)

238 MANIFEST_CHECK= \
239         @$(ECHO) "checking $<"; \
240         SVCCFG_DTD=$(SRC)/cmd/svc/dtd/service_bundle.dtd.1 \
241         SVCCFG_REPOSITORY=$(SRC)/cmd/svc/seed/global.db \
242         SVCCFG_CONFIGD_PATH=$(SRC)/cmd/svc/configd/svc.configd-native \
243         $(SRC)/cmd/svc/svccfg/svccfg-native validate $<

245 INS.file=       $(RM) $@; $(INS) -s -m $(FILEMODE) -f $(@D) $<
246 INS.dir=        $(INS) -s -d -m $(DIRMODE) $@
247 # installs and renames at once
248 #
249 INS.rename=     $(INS.file); $(MV) $(@D)/$(<F) $@

251 # install a link
252 INSLINKTARGET=  $<
253 INS.link=       $(RM) $@; $(LN) $(INSLINKTARGET) $@
254 INS.symlink=    $(RM) $@; $(SYMLINK) $(INSLINKTARGET) $@

256 #
257 # Python bakes the mtime of the .py file into the compiled .pyc and
258 # rebuilds if the baked-in mtime != the mtime of the source file
259 # (rather than only if it’s less than), thus when installing python
```

```
260 # files we must make certain to not adjust the mtime of the source
261 # (.py) file.
262 #
263 INS.pyfile=     $(RM) $@; $(SED) -e "1s:^\#!@PYTHON@:\#!$(PYTHON):" < $< > $@; $

265 # MACH must be set in the shell environment per uname -p on the build host
266 # More specific architecture variables should be set in lower makefiles.
267 #
268 # MACH64 is derived from MACH, and BUILD64 is set to '#' for
269 # architectures on which we do not build 64-bit versions.
270 # (There are no such architectures at the moment.)
271 #
272 # Set BUILD64=# in the environment to disable 64-bit amd64
273 # builds on i386 machines.

275 MACH64_1=       $(MACH:sparc=sparcv9)
276 MACH64=         $(MACH64_1:i386=amd64)

278 MACH32_1=       $(MACH:sparc=sparcv7)
279 MACH32=         $(MACH32_1:i386=i86)

281 sparc_BUILD64=
282 i386_BUILD64=
283 BUILD64=        $($(MACH)_BUILD64)

285 #
286 # C compiler mode. Future compilers may change the default on us,
287 # so force extended ANSI mode globally. Lower level makefiles can
288 # override this by setting CCMODE.
289 #
290 CCMODE=                 -Xa
291 CCMODE64=               -Xa

293 #
294 # C compiler verbose mode. This is so we can enable it globally,
295 # but turn it off in the lower level makefiles of things we cannot
296 # (or aren't going to) fix.
297 #
298 CCVERBOSE=              -v

300 # set this to the secret flag "-Wc,-Qiselect-v9abiwarn=1" to get warnings
301 # from the compiler about places the -xarch=v9 may differ from -xarch=v9c.
302 V9ABIWARN=

304 # set this to the secret flag "-Wc,-Qiselect-regsym=0" to disable register
305 # symbols (used to detect conflicts between objects that use global registers)
306 # we disable this now for safety, and because genunix doesn't link with
307 # this feature (the v9 default) enabled.
308 #
309 # REGSYM is separate since the C++ driver syntax is different.
310 CCREGSYM=               -Wc,-Qiselect-regsym=0
311 CCCREGSYM=              -Qoption cg -Qiselect-regsym=0

313 # Prevent the removal of static symbols by the SPARC code generator (cg).
314 # The x86 code generator (ube) does not remove such symbols and as such
315 # using this workaround is not applicable for x86.
316 #
317 CCSTATICSYM=            -Wc,-Qassembler-ounrefsym=0
318 #
319 # generate 32-bit addresses in the v9 kernel. Saves memory.
320 CCABS32=                -Wc,-xcode=abs32
321 #
322 # generate v9 code which tolerates callers using the v7 ABI, for the sake of
323 # system calls.
324 CC32BITCALLERS=         -_gcc=-massume-32bit-callers
```



```
326 # GCC, especially, is increasingly beginning to auto-inline functions and
327 # sadly does so separately not under the general -fno-inline-functions
328 # Additionally, we wish to prevent optimisations which cause GCC to clone
329 # functions -- in particular, these may cause unhelpful symbols to be
330 # emitted instead of function names
331 CCNOAUTOINLINE= -_gcc=-fno-inline-small-functions \
332         -_gcc=-fno-inline-functions-called-once \
333         -_gcc=-fno-ipa-cp

335 # One optimization the compiler might perform is to turn this:
336 #       #pragma weak foo
337 #       extern int foo;
338 #       if (&foo)
339 #               foo = 5;
340 # into
341 #       foo = 5;
342 # Since we do some of this (foo might be referenced in common kernel code
343 # but provided only for some cpu modules or platforms), we disable this
344 # optimization.
345 #
346 sparc_CCUNBOUND = -Wd,-xsafe=unboundsym
347 i386_CCUNBOUND  =
348 CCUNBOUND       = $($(MACH)_CCUNBOUND)

350 #
351 # compiler '-xarch' flag. This is here to centralize it and make it
352 # overridable for testing.
353 sparc_XARCH=    -m32
354 sparcv9_XARCH=  -m64
355 i386_XARCH=     -m32
356 amd64_XARCH=    -m64 -Ui386 -U__i386

358 # assembler '-xarch' flag.  Different from compiler '-xarch' flag.
359 sparc_AS_XARCH=         -xarch=v8plus
360 sparcv9_AS_XARCH=       -xarch=v9
361 i386_AS_XARCH=
362 amd64_AS_XARCH=         -xarch=amd64 -P -Ui386 -U__i386

364 #
365 # These flags define what we need to be 'standalone' i.e. -not- part
366 # of the rather more cosy userland environment.  This basically means
367 # the kernel.
368 #
369 # XX64  future versions of gcc will make -mcmodel=kernel imply -mno-red-zone
370 #
371 sparc_STAND_FLAGS=      -_gcc=-ffreestanding
372 sparcv9_STAND_FLAGS=    -_gcc=-ffreestanding
373 # Disabling MMX also disables 3DNow, disabling SSE also disables all later
374 # additions to SSE (SSE2, AVX ,etc.)
375 NO_SIMD=                -_gcc=-mno-mmx -_gcc=-mno-sse
376 i386_STAND_FLAGS=       -_gcc=-ffreestanding $(NO_SIMD)
377 amd64_STAND_FLAGS=      -xmodel=kernel $(NO_SIMD)

379 SAVEARGS=               -Wu,-save_args
380 amd64_STAND_FLAGS       += $(SAVEARGS)

382 STAND_FLAGS_32 = $($(MACH)_STAND_FLAGS)
383 STAND_FLAGS_64 = $($(MACH64)_STAND_FLAGS)

385 #
386 # disable the incremental linker
387 ILDOFF=                 -xildoff
388 #
389 XDEPEND=                -xdepend
390 XFFLAG=                 -xF=%all
391 XESS=                   -xs
```

```
392 XSTRCONST=              -xstrconst

394 #
395 # turn warnings into errors (C)
396 CERRWARN = -errtags=yes -errwarn=%all
397 CERRWARN += -erroff=E_EMPTY_TRANSLATION_UNIT
398 CERRWARN += -erroff=E_STATEMENT_NOT_REACHED

400 CERRWARN += -_gcc=-Wno-missing-braces
401 CERRWARN += -_gcc=-Wno-sign-compare
402 CERRWARN += -_gcc=-Wno-unknown-pragmas
403 CERRWARN += -_gcc=-Wno-unused-parameter
404 CERRWARN += -_gcc=-Wno-missing-field-initializers

406 # Unfortunately, this option can misfire very easily and unfixably.
407 CERRWARN +=     -_gcc=-Wno-array-bounds

409 # DEBUG v. -nd make for frequent unused variables, empty conditions, etc. in
410 # -nd builds
411 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-unused
412 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-empty-body

414 #
415 # turn warnings into errors (C++)
416 CCERRWARN=              -xwe

418 # C standard.  Keep Studio flags until we get rid of lint.
419 CSTD_GNU89=     -xc99=%none
420 CSTD_GNU99=     -xc99=%all
421 CSTD=           $(CSTD_GNU89)
422 C99LMODE=       $(CSTD:-xc99%=-Xc99%)

424 # In most places, assignments to these macros should be appended with +=
425 # (CPPFLAGS.first allows values to be prepended to CPPFLAGS).
426 sparc_CFLAGS=   $(sparc_XARCH) $(CCSTATICSYM)
427 sparcv9_CFLAGS= $(sparcv9_XARCH) -dalign $(CCVERBOSE) $(V9ABIWARN) $(CCREGSYM) \
428                 $(CCSTATICSYM)
429 i386_CFLAGS=    $(i386_XARCH)
430 amd64_CFLAGS=   $(amd64_XARCH)

432 sparc_ASFLAGS=  $(sparc_AS_XARCH)
433 sparcv9_ASFLAGS=$(sparcv9_AS_XARCH)
434 i386_ASFLAGS=   $(i386_AS_XARCH)
435 amd64_ASFLAGS=  $(amd64_AS_XARCH)

437 #
438 sparc_COPTFLAG=         -xO3
439 sparcv9_COPTFLAG=       -xO3
440 i386_COPTFLAG=          -O
441 amd64_COPTFLAG=         -xO3

443 COPTFLAG= $($(MACH)_COPTFLAG)
444 COPTFLAG64= $($(MACH64)_COPTFLAG)

446 # When -g is used, the compiler globalizes static objects
447 # (gives them a unique prefix). Disable that.
448 CNOGLOBAL= -W0,-noglobal

450 # Direct the Sun Studio compiler to use a static globalization prefix based on t
451 # name of the module rather than something unique. Otherwise, objects
452 # will not build deterministically, as subsequent compilations of identical
453 # source will yeild objects that always look different.
454 #
455 # In the same spirit, this will also remove the date from the N_OPT stab.
456 CGLOBALSTATIC= -W0,-xglobalstatic
```



```
458 # Sometimes we want all symbols and types in debugging information even
459 # if they aren't used.
460 CALLSYMS=       -W0,-xdbggen=no%usedonly

462 #
463 # Default debug format for Sun Studio 11 is dwarf, so force it to
464 # generate stabs.
465 #
466 DEBUGFORMAT=    -xdebugformat=stabs

468 #
469 # Flags used to build in debug mode for ctf generation.  Bugs in the Devpro
470 # compilers currently prevent us from building with cc-emitted DWARF.
471 #
472 CTF_FLAGS_sparc = -g -Wc,-Qiselect-T1 $(CSTD) $(CNOGLOBAL) $(CDWARFSTR)
473 CTF_FLAGS_i386  = -g $(CSTD) $(CNOGLOBAL) $(CDWARFSTR)

475 CTF_FLAGS_sparcv9       = $(CTF_FLAGS_sparc)
476 CTF_FLAGS_amd64         = $(CTF_FLAGS_i386)

478 # Sun Studio produces broken userland code when saving arguments.
479 $(__GNUC)CTF_FLAGS_amd64 += $(SAVEARGS)

481 CTF_FLAGS_32    = $(CTF_FLAGS_$(MACH)) $(DEBUGFORMAT)
482 CTF_FLAGS_64    = $(CTF_FLAGS_$(MACH64)) $(DEBUGFORMAT)
483 CTF_FLAGS       = $(CTF_FLAGS_32)

485 #
486 # Flags used with genoffsets
487 #
488 GOFLAGS = -_noecho \
489         $(CALLSYMS) \
490         $(CDWARFSTR)

492 OFFSETS_CREATE = $(GENOFFSETS) -s $(CTFSTABS) -r $(CTFCONVERT) \
493         $(CC) $(GOFLAGS) $(CFLAGS) $(CPPFLAGS)

495 OFFSETS_CREATE64 = $(GENOFFSETS) -s $(CTFSTABS) -r $(CTFCONVERT) \
496         $(CC) $(GOFLAGS) $(CFLAGS64) $(CPPFLAGS)

498 #
499 # tradeoff time for space (smaller is better)
500 #
501 sparc_SPACEFLAG         = -xspace -W0,-Lt
502 sparcv9_SPACEFLAG       = -xspace -W0,-Lt
503 i386_SPACEFLAG          = -xspace
504 amd64_SPACEFLAG         =

506 SPACEFLAG               = $($(MACH)_SPACEFLAG)
507 SPACEFLAG64             = $($(MACH64)_SPACEFLAG)

509 #
510 # The Sun Studio 11 compiler has changed the behaviour of integer
511 # wrap arounds and so a flag is needed to use the legacy behaviour
512 # (without this flag panics/hangs could be exposed within the source).
513 #
514 sparc_IROPTFLAG         = -W2,-xwrap_int
515 sparcv9_IROPTFLAG       = -W2,-xwrap_int
516 i386_IROPTFLAG          =
517 amd64_IROPTFLAG         =

519 IROPTFLAG               = $($(MACH)_IROPTFLAG)
520 IROPTFLAG64             = $($(MACH64)_IROPTFLAG)

522 sparc_XREGSFLAG         = -xregs=no%appl
523 sparcv9_XREGSFLAG       = -xregs=no%appl
```

```
524 i386_XREGSFLAG          =
525 amd64_XREGSFLAG         =

527 XREGSFLAG               = $($(MACH)_XREGSFLAG)
528 XREGSFLAG64             = $($(MACH64)_XREGSFLAG)

530 # dmake SOURCEDEBUG=yes ... enables source-level debugging information, and
531 # avoids stripping it.
532 SOURCEDEBUG     = $(POUND_SIGN)
533 SRCDBGBLD       = $(SOURCEDEBUG:yes=)

535 #
536 # These variables are intended ONLY for use by developers to safely pass extra
537 # flags to the compilers without unintentionally overriding Makefile-set
538 # flags.  They should NEVER be set to any value in a Makefile.
539 #
540 # They come last in the associated FLAGS variable such that they can
541 # explicitly override things if necessary, there are gaps in this, but it's
542 # the best we can manage.
543 #
544 CUSERFLAGS              =
545 CUSERFLAGS64            = $(CUSERFLAGS)
546 CCUSERFLAGS             =
547 CCUSERFLAGS64           = $(CCUSERFLAGS)

549 CSOURCEDEBUGFLAGS       =
550 CCSOURCEDEBUGFLAGS      =
551 $(SRCDBGBLD)CSOURCEDEBUGFLAGS   = -g -xs
552 $(SRCDBGBLD)CCSOURCEDEBUGFLAGS  = -g -xs

554 CFLAGS=         $(COPTFLAG) $($(MACH)_CFLAGS) $(SPACEFLAG) $(CCMODE) \
555                 $(ILDOFF) $(CERRWARN) $(CSTD) $(CCUNBOUND) $(IROPTFLAG) \
556                 $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
557                 $(CUSERFLAGS)
558 CFLAGS64=       $(COPTFLAG64) $($(MACH64)_CFLAGS) $(SPACEFLAG64) $(CCMODE64) \
559                 $(ILDOFF) $(CERRWARN) $(CSTD) $(CCUNBOUND) $(IROPTFLAG64) \
560                 $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
561                 $(CUSERFLAGS64)
562 #
563 # Flags that are used to build parts of the code that are subsequently
564 # run on the build machine (also known as the NATIVE_BUILD).
565 #
566 NATIVE_CFLAGS=  $(COPTFLAG) $($(NATIVE_MACH)_CFLAGS) $(CCMODE) \
567                 $(ILDOFF) $(CERRWARN) $(CSTD) $($(NATIVE_MACH)_CCUNBOUND) \
568                 $(IROPTFLAG) $(CGLOBALSTATIC) $(CCNOAUTOINLINE) \
569                 $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS)

571 DTEXTDOM=-DTEXT_DOMAIN=\"$(TEXT_DOMAIN)\"       # For messaging.
572 DTS_ERRNO=-D_TS_ERRNO
573 CPPFLAGS.first= # Please keep empty.  Only lower makefiles should set this.
574 CPPFLAGS.master=$(DTEXTDOM) $(DTS_ERRNO) \
575         $(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) $(ENVCPPFLAGS4) \
576         $(ADJUNCT_PROTO:%=-I%/usr/include)
577 CPPFLAGS.native=$(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) \
578                 $(ENVCPPFLAGS4) -I$(NATIVE_ADJUNCT)/include
579 CPPFLAGS=       $(CPPFLAGS.first) $(CPPFLAGS.master)
580 AS_CPPFLAGS=    $(CPPFLAGS.first) $(CPPFLAGS.master)
581 JAVAFLAGS=      -source 1.6 -target 1.6 -Xlint:deprecation,-options

583 #
584 # For source message catalogue
585 #
586 .SUFFIXES: $(SUFFIXES) .i .po
587 MSGROOT= $(ROOT)/catalog
588 MSGDOMAIN= $(MSGROOT)/$(TEXT_DOMAIN)
589 MSGDOMAINPOFILE = $(MSGDOMAIN)/$(POFILE)
```



```
590 DCMSGDOMAIN= $(MSGROOT)/LC_TIME/$(TEXT_DOMAIN)
591 DCMSGDOMAINPOFILE = $(DCMSGDOMAIN)/$(DCFILE:.dc=.po)

593 CLOBBERFILES += $(POFILE) $(POFILES)
594 COMPILE.cpp= $(CC) -E -C $(CFLAGS) $(CPPFLAGS)
595 XGETTEXT= /usr/bin/xgettext
596 XGETFLAGS= -c TRANSLATION_NOTE
597 GNUXGETTEXT= /usr/gnu/bin/xgettext
598 GNUXGETFLAGS= --add-comments=TRANSLATION_NOTE --keyword=_ \
599         --strict --no-location --omit-header
600 BUILD.po= $(XGETTEXT) $(XGETFLAGS) -d $(<F) $<.i ;\
601         $(RM)   $@ ;\
602         $(SED) "/^domain/d" < $(<F).po > $@ ;\
603         $(RM) $(<F).po $<.i

605 #
606 # This is overwritten by local Makefile when PROG is a list.
607 #
608 POFILE= $(PROG).po

610 sparc_CCFLAGS=          -cg92 -compat=4 \
611                         -Qoption ccfe -messages=no%anachronism \
612                         $(CCERRWARN)
613 sparcv9_CCFLAGS=        $(sparcv9_XARCH) -dalign -compat=5 \
614                         -Qoption ccfe -messages=no%anachronism \
615                         -Qoption ccfe -features=no%conststrings \
616                         $(CCCREGSYM) \
617                         $(CCERRWARN)
618 i386_CCFLAGS=           -compat=4 \
619                         -Qoption ccfe -messages=no%anachronism \
620                         -Qoption ccfe -features=no%conststrings \
621                         $(CCERRWARN)
622 amd64_CCFLAGS=          $(amd64_XARCH) -compat=5 \
623                         -Qoption ccfe -messages=no%anachronism \
624                         -Qoption ccfe -features=no%conststrings \
625                         $(CCERRWARN)

627 sparc_CCOPTFLAG=        -O
628 sparcv9_CCOPTFLAG=      -O
629 i386_CCOPTFLAG=         -O
630 amd64_CCOPTFLAG=        -O

632 CCOPTFLAG=      $($(MACH)_CCOPTFLAG)
633 CCOPTFLAG64=    $($(MACH64)_CCOPTFLAG)
634 CCFLAGS=        $(CCOPTFLAG) $($(MACH)_CCFLAGS) $(CCSOURCEDEBUGFLAGS) \
635                 $(CCUSERFLAGS)
636 CCFLAGS64=      $(CCOPTFLAG64) $($(MACH64)_CCFLAGS) $(CCSOURCEDEBUGFLAGS) \
637                 $(CCUSERFLAGS64)

639 #
640 #
641 #
642 ELFWRAP_FLAGS   =
643 ELFWRAP_FLAGS64 =       -64

645 #
646 # Various mapfiles that are used throughout the build, and delivered to
647 # /usr/lib/ld.
648 #
649 MAPFILE.NED_i386 =      $(SRC)/common/mapfiles/common/map.noexdata
650 MAPFILE.NED_sparc =
651 MAPFILE.NED =           $(MAPFILE.NED_$(MACH))
652 MAPFILE.PGA =           $(SRC)/common/mapfiles/common/map.pagealign
653 MAPFILE.NES =           $(SRC)/common/mapfiles/common/map.noexstk
654 MAPFILE.FLT =           $(SRC)/common/mapfiles/common/map.filter
655 MAPFILE.LEX =           $(SRC)/common/mapfiles/common/map.lex.yy
```

```
#
# Generated mapfiles that are compiler specific, and used throughout the
# build.  These mapfiles are not delivered in /usr/lib/ld.
#
MAPFILE.NGB_sparc=      $(SRC)/common/mapfiles/gen/sparc_cc_map.noexeglobs
$(__GNUC64)MAPFILE.NGB_sparc= \
                        $(SRC)/common/mapfiles/gen/sparc_gcc_map.noexeglobs
MAPFILE.NGB_sparcv9=    $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexeglobs
$(__GNUC64)MAPFILE.NGB_sparcv9= \
                        $(SRC)/common/mapfiles/gen/sparcv9_gcc_map.noexeglobs
MAPFILE.NGB_i386=       $(SRC)/common/mapfiles/gen/i386_cc_map.noexeglobs
$(__GNUC64)MAPFILE.NGB_i386= \
                        $(SRC)/common/mapfiles/gen/i386_gcc_map.noexeglobs
MAPFILE.NGB_amd64=      $(SRC)/common/mapfiles/gen/amd64_cc_map.noexeglobs
$(__GNUC64)MAPFILE.NGB_amd64= \
                        $(SRC)/common/mapfiles/gen/amd64_gcc_map.noexeglobs
MAPFILE.NGB =           $(MAPFILE.NGB_$(MACH))

#
# A generic interface mapfile name, used by various dynamic objects to define
# the interfaces and interposers the object must export.
#
MAPFILE.INT =           mapfile-intf

#
# LDLIBS32 and LDLIBS64 can be set in the environment to override the following
# assignments.
#
# These environment settings make sure that no libraries are searched outside
# of the local workspace proto area:
#       LDLIBS32=-YP,$ROOT/lib:$ROOT/usr/lib
#       LDLIBS64=-YP,$ROOT/lib/$MACH64:$ROOT/usr/lib/$MACH64
#
LDLIBS32 =      $(ENVLDLIBS1) $(ENVLDLIBS2) $(ENVLDLIBS3)
LDLIBS32 +=     $(ADJUNCT_PROTO:%=-L%/usr/lib -L%/lib)
LDLIBS.cmd =    $(LDLIBS32)
LDLIBS.lib =    $(LDLIBS32)

LDLIBS64 =      $(ENVLDLIBS1:%=%/$(MACH64)) \
                $(ENVLDLIBS2:%=%/$(MACH64)) \
                $(ENVLDLIBS3:%=%/$(MACH64))
LDLIBS64 +=     $(ADJUNCT_PROTO:%=-L%/usr/lib/$(MACH64) -L%/lib/$(MACH64))

#
# Define compilation macros.
#
COMPILE.c=      $(CC) $(CFLAGS) $(CPPFLAGS) -c
COMPILE64.c=    $(CC) $(CFLAGS64) $(CPPFLAGS) -c
COMPILE.cc=     $(CCC) $(CCFLAGS) $(CPPFLAGS) -c
COMPILE64.cc=   $(CCC) $(CCFLAGS64) $(CPPFLAGS) -c
COMPILE.s=      $(AS) $(ASFLAGS) $(AS_CPPFLAGS)
COMPILE64.s=    $(AS) $(ASFLAGS) $($(MACH64)_AS_XARCH) $(AS_CPPFLAGS)
COMPILE.d=      $(DTRACE) -G -32
COMPILE64.d=    $(DTRACE) -G -64
COMPILE.b=      $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))
COMPILE64.b=    $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))

CLASSPATH=      .
COMPILE.java=   $(JAVAC) $(JAVAFLAGS) -classpath $(CLASSPATH)

#
# Link time macros
#
CCNEEDED                = -lC
CCEXTNEEDED             = -lCrun -lCstd
```



```
$(__GNUC)CCNEEDED       = -L$(GCCLIBDIR) -lstdc++ -lgcc_s
$(__GNUC)CCEXTNEEDED    = $(CCNEEDED)

LINK.c=         $(CC) $(CFLAGS) $(CPPFLAGS) $(LDFLAGS)
LINK64.c=       $(CC) $(CFLAGS64) $(CPPFLAGS) $(LDFLAGS)
NORUNPATH=      -norunpath -nolib
LINK.cc=        $(CCC) $(CCFLAGS) $(CPPFLAGS) $(NORUNPATH) \
                $(LDFLAGS) $(CCNEEDED)
LINK64.cc=      $(CCC) $(CCFLAGS64) $(CPPFLAGS) $(NORUNPATH) \
                $(LDFLAGS) $(CCNEEDED)

#
# lint macros
#
# Note that the undefine of __PRAGMA_REDEFINE_EXTNAME can be removed once
# ON is built with a version of lint that has the fix for 4484186.
#
ALWAYS_LINT_DEFS =      -errtags=yes -s
ALWAYS_LINT_DEFS +=     -erroff=E_PTRDIFF_OVERFLOW
ALWAYS_LINT_DEFS +=     -erroff=E_ASSIGN_NARROW_CONV
ALWAYS_LINT_DEFS +=     -U__PRAGMA_REDEFINE_EXTNAME
ALWAYS_LINT_DEFS +=     $(C99LMODE)
ALWAYS_LINT_DEFS +=     -errsecurity=$(SECLEVEL)
ALWAYS_LINT_DEFS +=     -erroff=E_SEC_CREAT_WITHOUT_EXCL
ALWAYS_LINT_DEFS +=     -erroff=E_SEC_FORBIDDEN_WARN_CREAT
# XX64 -- really only needed for amd64 lint
ALWAYS_LINT_DEFS +=     -erroff=E_ASSIGN_INT_TO_SMALL_INT
ALWAYS_LINT_DEFS +=     -erroff=E_CAST_INT_CONST_TO_SMALL_INT
ALWAYS_LINT_DEFS +=     -erroff=E_CAST_INT_TO_SMALL_INT
ALWAYS_LINT_DEFS +=     -erroff=E_CAST_TO_PTR_FROM_INT
ALWAYS_LINT_DEFS +=     -erroff=E_COMP_INT_WITH_LARGE_INT
ALWAYS_LINT_DEFS +=     -erroff=E_INTEGRAL_CONST_EXP_EXPECTED
ALWAYS_LINT_DEFS +=     -erroff=E_PASS_INT_TO_SMALL_INT
ALWAYS_LINT_DEFS +=     -erroff=E_PTR_CONV_LOSES_BITS

# This forces lint to pick up note.h and sys/note.h from Devpro rather than
# from the proto area.  The note.h that ON delivers would disable NOTE().
ONLY_LINT_DEFS =        -I$(SPRO_VROOT)/prod/include/lint

SECLEVEL=       core
LINT.c=         $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS) $(CPPFLAGS) \
                $(ALWAYS_LINT_DEFS)
LINT64.c=       $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS64) $(CPPFLAGS) \
                $(ALWAYS_LINT_DEFS)
LINT.s=         $(LINT.c)

# For some future builds, NATIVE_MACH and MACH might be different.
# Therefore, NATIVE_MACH needs to be redefined in the
# environment as ‘uname -p‘ to override this macro.
#
# For now at least, we cross-compile amd64 on i386 machines.
NATIVE_MACH=    $(MACH:amd64=i386)

# Define native compilation macros
#

# Base directory where compilers are loaded.
# Defined here so it can be overridden by developer.
#
SPRO_ROOT=              $(BUILD_TOOLS)/SUNWspro
SPRO_VROOT=             $(SPRO_ROOT)/SS12
GNU_ROOT=               /usr

# Till SS12u1 formally becomes the NV CBE, LINT is hard
# coded to be picked up from the $SPRO_ROOT/sunstudio12.1/
# location. Impacted variables are sparc_LINT, sparcv9_LINT,
```

```
788 # i386_LINT, amd64_LINT.
789 # Reset them when SS12u1 is rolled out.
790 #

792 # Specify platform compiler versions for languages
793 # that we use (currently only c and c++).
794 #
795 sparc_CC=               $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
796 $(__GNUC)sparc_CC=      $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
797 sparc_CCC=              $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
798 $(__GNUC)sparc_CCC=     $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
799 sparc_CPP=              /usr/ccs/lib/cpp
800 sparc_AS=               /usr/ccs/bin/as -xregsym=no
801 sparc_LD=               /usr/ccs/bin/ld
802 sparc_LINT=             $(SPRO_ROOT)/sunstudio12.1/bin/lint

804 sparcv9_CC=            $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
805 $(__GNUC64)sparcv9_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
806 sparcv9_CCC=           $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
807 $(__GNUC64)sparcv9_CCC=$(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
808 sparcv9_CPP=           /usr/ccs/lib/cpp
809 sparcv9_AS=            /usr/ccs/bin/as -xregsym=no
810 sparcv9_LD=            /usr/ccs/bin/ld
811 sparcv9_LINT=          $(SPRO_ROOT)/sunstudio12.1/bin/lint

813 i386_CC=              $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
814 $(__GNUC)i386_CC=     $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
815 i386_CCC=             $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
816 $(__GNUC)i386_CCC=    $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
817 i386_CPP=             /usr/ccs/lib/cpp
818 i386_AS=              /usr/ccs/bin/as
819 $(__GNUC)i386_AS=     $(ONBLD_TOOLS)/bin/$(MACH)/aw
820 i386_LD=              /usr/ccs/bin/ld
821 i386_LINT=            $(SPRO_ROOT)/sunstudio12.1/bin/lint

823 amd64_CC=             $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
824 $(__GNUC64)amd64_CC=  $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
825 amd64_CCC=            $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
826 $(__GNUC64)amd64_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
827 amd64_CPP=            /usr/ccs/lib/cpp
828 amd64_AS=             $(ONBLD_TOOLS)/bin/$(MACH)/aw
829 amd64_LD=             /usr/ccs/bin/ld
830 amd64_LINT=           $(SPRO_ROOT)/sunstudio12.1/bin/lint

832 NATIVECC=             $($(NATIVE_MACH)_CC)
833 NATIVECCC=            $($(NATIVE_MACH)_CCC)
834 NATIVECPP=            $($(NATIVE_MACH)_CPP)
835 NATIVEAS=             $($(NATIVE_MACH)_AS)
836 NATIVELD=             $($(NATIVE_MACH)_LD)
837 NATIVELINT=           $($(NATIVE_MACH)_LINT)

839 #
840 # Makefile.master.64 overrides these settings
841 #
842 CC=                   $(NATIVECC)
843 CCC=                  $(NATIVECCC)
844 CPP=                  $(NATIVECPP)
845 AS=                   $(NATIVEAS)
846 LD=                   $(NATIVELD)
847 LINT=                 $(NATIVELINT)

849 # The real compilers used for this build
850 CW_CC_CMD=            $(CC) -_compiler
851 CW_CCC_CMD=           $(CCC) -_compiler
852 REAL_CC=              $(CW_CC_CMD:sh)
853 REAL_CCC=             $(CW_CCC_CMD:sh)
```



```
855 # Pass -Y flag to cpp (method of which is release-dependent)
856 CCYFLAG=                -Y I,

858 BDIRECT=        -Bdirect
859 BDYNAMIC=       -Bdynamic
860 BLOCAL=         -Blocal
861 BNODIRECT=      -Bnodirect
862 BREDUCE=        -Breduce
863 BSTATIC=        -Bstatic

865 ZDEFS=          -zdefs
866 ZDIRECT=        -zdirect
867 ZIGNORE=        -zignore
868 ZINITFIRST=     -zinitfirst
869 ZINTERPOSE=     -zinterpose
870 ZLAZYLOAD=      -zlazyload
871 ZLOADFLTR=      -zloadfltr
872 ZMULDEFS=       -zmuldefs
873 ZNODEFAULTLIB=  -znodefaultlib
874 ZNODEFS=        -znodefs
875 ZNODELETE=      -znodelete
876 ZNODLOPEN=      -znodlopen
877 ZNODUMP=        -znodump
878 ZNOLAZYLOAD=    -znolazyload
879 ZNOLDYNSYM=     -znoldynsym
880 ZNORELOC=       -znoreloc
881 ZNOVERSION=     -znoversion
882 ZRECORD=        -zrecord
883 ZREDLOCSYM=     -zredlocsym
884 ZTEXT=          -ztext
885 ZVERBOSE=       -zverbose

887 GSHARED=        -G
888 CCMT=           -mt

890 # Handle different PIC models on different ISAs
891 # (May be overridden by lower-level Makefiles)

893 sparc_C_PICFLAGS =      -K pic
894 sparcv9_C_PICFLAGS =    -K pic
895 i386_C_PICFLAGS =       -K pic
896 amd64_C_PICFLAGS =      -K pic
897 C_PICFLAGS =            $($(MACH)_C_PICFLAGS)
898 C_PICFLAGS64 =          $($(MACH64)_C_PICFLAGS)

900 sparc_C_BIGPICFLAGS =   -K PIC
901 sparcv9_C_BIGPICFLAGS = -K PIC
902 i386_C_BIGPICFLAGS =    -K PIC
903 amd64_C_BIGPICFLAGS =   -K PIC
904 C_BIGPICFLAGS =         $($(MACH)_C_BIGPICFLAGS)
905 C_BIGPICFLAGS64 =       $($(MACH64)_C_BIGPICFLAGS)

907 # CC requires there to be no space between '-K' and 'pic' or 'PIC'.
908 sparc_CC_PICFLAGS =     -Kpic
909 sparcv9_CC_PICFLAGS =   -KPIC
910 i386_CC_PICFLAGS =      -Kpic
911 amd64_CC_PICFLAGS =     -Kpic
912 CC_PICFLAGS =           $($(MACH)_CC_PICFLAGS)
913 CC_PICFLAGS64 =         $($(MACH64)_CC_PICFLAGS)

915 AS_PICFLAGS=            $(C_PICFLAGS)
916 AS_BIGPICFLAGS=         $(C_BIGPICFLAGS)

918 #
919 # Default label for CTF sections
```

```
#
CTFCVTFLAGS=            -i -L VERSION

#
# Override to pass module-specific flags to ctfmerge.  Currently used only by
# krtld to turn on fuzzy matching, and source-level debugging to inhibit
# stripping.
#
CTFMRGFLAGS=

CTFCONVERT_O            = $(CTFCONVERT) $(CTFCVTFLAGS) $@

# Rules (normally from make.rules) and macros which are used for post
# processing files. Normally, these do stripping of the comment section
# automatically.
#    RELEASE_CM:        Should be editted to reflect the release.
#    POST_PROCESS_O:    Post-processing for '.o' files.
#    POST_PROCESS_A:    Post-processing for '.a' files (currently null).
#    POST_PROCESS_SO:   Post-processing for '.so' files.
#    POST_PROCESS:      Post-processing for executable files (no suffix).
# Note that these macros are not completely generalized as they are to be
# used with the file name to be processed following.
#
# It is left as an exercise to Release Engineering to embellish the generation
# of the release comment string.
#
#       If this is a standard development build:
#               compress the comment section (mcs -c)
#               add the standard comment (mcs -a $(RELEASE_CM))
#               add the development specific comment (mcs -a $(DEV_CM))
#
#       If this is an installation build:
#               delete the comment section (mcs -d)
#               add the standard comment (mcs -a $(RELEASE_CM))
#               add the development specific comment (mcs -a $(DEV_CM))
#
#       If this is an release build:
#               delete the comment section (mcs -d)
#               add the standard comment (mcs -a $(RELEASE_CM))
#
# The following list of macros are used in the definition of RELEASE_CM
# which is used to label all binaries in the build:
#
#       RELEASE         Specific release of the build, eg: 5.2
#       RELEASE_MAJOR   Major version number part of $(RELEASE)
#       RELEASE_MINOR   Minor version number part of $(RELEASE)
#       VERSION         Version of the build (alpha, beta, Generic)
#       PATCHID         If this is a patch this value should contain
#                       the patchid value (eg: "Generic 100832-01"), otherwise
#                       it will be set to $(VERSION)
#       RELEASE_DATE    Date of the Release Build
#       PATCH_DATE      Date the patch was created, if this is blank it
#                       will default to the RELEASE_DATE
#
RELEASE_MAJOR=  5
RELEASE_MINOR=  11
RELEASE=        $(RELEASE_MAJOR).$(RELEASE_MINOR)
VERSION=        SunOS Development
PATCHID=        $(VERSION)
RELEASE_DATE=   release date not set
PATCH_DATE=     $(RELEASE_DATE)
RELEASE_CM=     "@($(POUND_SIGN))SunOS $(RELEASE) $(PATCHID) $(PATCH_DATE)"
DEV_CM=         "@($(POUND_SIGN))SunOS Internal Development: non-nightly build"

PROCESS_COMMENT=  @?${MCS} -d -a $(RELEASE_CM) -a $(DEV_CM)
$(RELEASE_BUILD)PROCESS_COMMENT=   @?${MCS} -d -a $(RELEASE_CM)
```



```
STRIP_STABS=                        $(STRIP) -x $@
$(SRCDBGBLD)STRIP_STABS=            :

POST_PROCESS_O=
POST_PROCESS_A=
POST_PROCESS_SO=        $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
                        $(ELFSIGN_OBJECT)
POST_PROCESS=           $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
                        $(ELFSIGN_OBJECT)

#
# chk4ubin is a tool that inspects a module for a symbol table
# ELF section size which can trigger an OBP bug on older platforms.
# This problem affects only specific sun4u bootable modules.
#
CHK4UBIN=       $(ONBLD_TOOLS)/bin/$(MACH)/chk4ubin
CHK4UBINFLAGS=
CHK4UBINARY=    $(CHK4UBIN) $(CHK4UBINFLAGS) $@

#
# PKGARCHIVE specifies the default location where packages should be
# placed if built.
#
$(RELEASE_BUILD)PKGARCHIVESUFFIX=       -nd
PKGARCHIVE=$(SRC)/../../packages/$(MACH)/nightly$(PKGARCHIVESUFFIX)

#
# The repositories will be created with these publisher settings.  To
# update an image to the resulting repositories, this must match the
# publisher name provided to "pkg set-publisher."
#
PKGPUBLISHER_REDIST=    on-nightly
PKGPUBLISHER_NONREDIST= on-extra

#       Default build rules which perform comment section post-processing.
#
.c:
        $(LINK.c) -o $@ $< $(LDLIBS)
        $(POST_PROCESS)
.c.o:
        $(COMPILE.c) $(OUTPUT_OPTION) $< $(CTFCONVERT_HOOK)
        $(POST_PROCESS_O)
.c.a:
        $(COMPILE.c) -o $% $<
        $(PROCESS_COMMENT) $%
        $(AR) $(ARFLAGS) $@ $%
        $(RM) $%
.s.o:
        $(COMPILE.s) -o $@ $<
        $(POST_PROCESS_O)
.s.a:
        $(COMPILE.s) -o $% $<
        $(PROCESS_COMMENT) $%
        $(AR) $(ARFLAGS) $@ $%
        $(RM) $%
.cc:
        $(LINK.cc) -o $@ $< $(LDLIBS)
        $(POST_PROCESS)
.cc.o:
        $(COMPILE.cc) $(OUTPUT_OPTION) $<
        $(POST_PROCESS_O)
.cc.a:
        $(COMPILE.cc) -o $% $<
        $(AR) $(ARFLAGS) $@ $%
        $(PROCESS_COMMENT) $%
```

```
1052         $(RM) $%
1053 .y:
1054         $(YACC.y) $<
1055         $(LINK.c) -o $@ y.tab.c $(LDLIBS)
1056         $(POST_PROCESS)
1057         $(RM) y.tab.c
1058 .y.o:
1059         $(YACC.y) $<
1060         $(COMPILE.c) -o $@ y.tab.c $(CTFCONVERT_HOOK)
1061         $(POST_PROCESS_O)
1062         $(RM) y.tab.c
1063 .l:
1064         $(RM) $*.c
1065         $(LEX.l) $< > $*.c
1066         $(LINK.c) -o $@ $*.c -ll $(LDLIBS)
1067         $(POST_PROCESS)
1068         $(RM) $*.c
1069 .l.o:
1070         $(RM) $*.c
1071         $(LEX.l) $< > $*.c
1072         $(COMPILE.c) -o $@ $*.c $(CTFCONVERT_HOOK)
1073         $(POST_PROCESS_O)
1074         $(RM) $*.c

1076 .bin.o:
1077         $(COMPILE.b) -o $@ $<
1078         $(POST_PROCESS_O)

1080 .java.class:
1081         $(COMPILE.java) $<

1083 # Bourne and Korn shell script message catalog build rules.
1084 # We extract all gettext strings with sed(1) (being careful to permit
1085 # multiple gettext strings on the same line), weed out the dups, and
1086 # build the catalogue with awk(1).

1088 .sh.po .ksh.po:
1089         $(SED) -n -e ":a"                                   \
1090                   -e "h"                                        \
1091                   -e "s/.*gettext *\(\"[^\"]*\"\).*/\1/p"       \
1092                   -e "x"                                        \
1093                   -e "s/\(.*\)gettext *\"[^\"]*\"\(.*\)/\1\2/"  \
1094                   -e "t a"                                      \
1095                 $< | sort -u | $(AWK) '{ print "msgid\t" $$0 "\nmsgstr" }' > $@

1097 #
1098 # Python and Perl executable and message catalog build rules.
1099 #
1100 .SUFFIXES: .pl .pm .py .pyc

1102 .pl:
1103         $(RM) $@;
1104         $(SED) -e "s@TEXT_DOMAIN@\"$(TEXT_DOMAIN)\"@" $< > $@;
1105         $(CHMOD) +x $@

1107 .py:
1108         $(RM) $@; $(SED) -e "1s:^\#!@PYTHON@:\#!$(PYTHON):" < $< > $@; $(CHMOD)

1110 .py.pyc:
1111         $(RM) $@
1112         $(PYTHON) -mpy_compile $<
1113         @[ $(<)c = $@ ] || $(MV) $(<)c $@

1115 .py.po:
1116         $(GNUXGETTEXT) $(GNUXGETFLAGS) -d $(<F:%.py=%) $< ;
```



```
1118 .pl.po .pm.po:
1119         $(XGETTEXT) $(XGETFLAGS) -d $(<F) $< ;
1120         $(RM)   $@ ;
1121         $(SED) "/^domain/d" < $(<F).po > $@ ;
1122         $(RM) $(<F).po

1124 #
1125 # When using xgettext, we want messages to go to the default domain,
1126 # rather than the specified one.  This special version of the
1127 # COMPILE.cpp macro effectively prevents expansion of TEXT_DOMAIN,
1128 # causing xgettext to put all messages into the default domain.
1129 #
1130 CPPFORPO=$(COMPILE.cpp:\"$(TEXT_DOMAIN)\"=TEXT_DOMAIN)

1132 .c.i:
1133         $(CPPFORPO) $< > $@

1135 .h.i:
1136         $(CPPFORPO) $< > $@

1138 .y.i:
1139         $(YACC) -d $<
1140         $(CPPFORPO) y.tab.c  > $@
1141         $(RM) y.tab.c

1143 .l.i:
1144         $(LEX) $<
1145         $(CPPFORPO) lex.yy.c  > $@
1146         $(RM) lex.yy.c

1148 .c.po:
1149         $(CPPFORPO) $< > $<.i
1150         $(BUILD.po)

1152 .cc.po:
1153         $(CPPFORPO) $< > $<.i
1154         $(BUILD.po)

1156 .y.po:
1157         $(YACC) -d $<
1158         $(CPPFORPO) y.tab.c  > $<.i
1159         $(BUILD.po)
1160         $(RM) y.tab.c

1162 .l.po:
1163         $(LEX) $<
1164         $(CPPFORPO) lex.yy.c  > $<.i
1165         $(BUILD.po)
1166         $(RM) lex.yy.c

1168 #
1169 # Rules to perform stylistic checks
1170 #
1171 .SUFFIXES: .x .xml .check .xmlchk

1173 .h.check:
1174         $(DOT_H_CHECK)

1176 .x.check:
1177         $(DOT_X_CHECK)

1179 .xml.xmlchk:
1180         $(MANIFEST_CHECK)

1182 #
1183 # Include rules to render automated sccs get rules "safe".
```

```
1184 #
1185 include $(SRC)/Makefile.noget
```

```
**********************************************************
   14513 Sun Mar 18 01:12:57 2018
new/usr/src/lib/Makefile
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

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

  22 #
  23 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
  24 # Copyright (c) 2012 by Delphix. All rights reserved.
  25 # Copyright (c) 2012, Joyent, Inc. All rights reserved.
  26 # Copyright (c) 2013 Gary Mills
  27 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
  28 # Copyright (c) 2015 Gary Mills
  29 # Copyright 2016 Toomas Soome <tsoome@me.com>
  30 # Copyright 2018 Nexenta Systems, Inc.
  31 # Copyright (c) 2016, Chris Fraire <cfraire@me.com>.
  32 # Copyright 2017 RackTop Systems.
  33 #

  35 include ../Makefile.master

  37 #       Note that libcurses installs commands along with its library.
  38 #       This is a minor bug which probably should be fixed.
  39 #       Note also that a few extra libraries are kept in cmd source.
  40 #
  41 # Certain libraries are linked with, hence depend on, other libraries.
  42 # Library dependencies are called out explicitly, see "Library
  43 # interdependencies" below.
  44 .PARALLEL:

  46 # Build libc and its dependencies
  47 SUBDIRS=                        \
  48         crt                    \
  49         ../cmd/sgs/libconv     \
  50         ../cmd/sgs/libdl       \
  51         libc                   \
  52         .WAIT

  54 # Build libraries with lots of dependents
  55 SUBDIRS +=              \
  56         libm           \
  57         libmd          \
  58         libmp          \
  59         libnsl         \
  60         libnvpair      \
  61         libsocket      \
```



```
  62         .WAIT

  64 # Build everything else. Aside from explicit dependencies, all libraries
  65 # are built in parallel.
  66 SUBDIRS +=                              \
  67         ../cmd/sendmail/libmilter       \
  68         ../cmd/sgs/libelf       \
  69         ../cmd/sgs/librtld_db   \
  70         abi                     \
  71         auditd_plugins  \
  72         brand           \
  73         c_synonyms      \
  74         cfgadm_plugins  \
  75         crypt_modules   \
  76         extendedFILE    \
  77         fm              \
  78         getloginx       \
  79         gss_mechs/mech_dh       \
  80         gss_mechs/mech_dummy    \
  81         gss_mechs/mech_krb5     \
  82         gss_mechs/mech_spnego   \
  83         hal             \
  84         hbaapi          \
  85         krb5            \
  86         libadm          \
  87         libads          \
  88         libadt_jni      \
  89         libadutils      \
  90         libaio          \
  91         libast          \
  92         libavl          \
  93         libbe           \
  94         libbrand        \
  95         libbsdmalloc    \
  96         libbsm          \
  97         libc_db         \
  98         libcfgadm       \
  99         libcmd          \
 100         libcmdutils     \
 101         libcommputil    \
 102         libcontract     \
 103         libcpc          \
 104         libcrypt        \
 105         libcryptoutil   \
 106         libctf          \
 107         libcurses       \
 108         libdevice       \
 109         libdevid        \
 110         libdevinfo      \
 111         libdhcpagent    \
 112         libdhcputil     \
 113         libdisasm       \
 114         libdiskmgt      \
 115         libdladm        \
 116         libdll          \
 117         libdlpi         \
 118         libdns_sd       \
 119         libdoor         \
 120         libdscfg        \
 121         libdtrace       \
 122         libdtrace_jni   \
 123         libefi          \
 124         libelfsign      \
 125         libeti          \
 126         libexacct       \
 127         libexacct/demo  \
```

```
128         libfakekernel   \
129         libfcoe         \
130         libficl         \
131         libfru          \
132         libfruutils     \
133         libfsmgt        \
134         libfstyp        \
135         libgen          \
136         libgrubmgmt     \
137         libgss          \
138         libhotplug      \
139         libidmap        \
140         libilb          \
141         libima          \
142         libinetsvc      \
143         libinetutil     \
144         libinstzones    \
145         libintl         \
146         libipadm        \
147         libipd          \
148         libipmi         \
149         libipmp         \
150         libipp          \
151         libipsecutil    \
152         libiscsit       \
153         libkmf          \
154         libkrb5         \
155         libkstat        \
156         libkvm          \
157         libldap5        \
158         liblgrp         \
159         liblm           \
160         libm1           \
161         libmail         \
162         libmalloc       \
163         libmapid        \
164         libmapmalloc    \
165         libmd5          \
166         libmlrpc        \
167         libmtmalloc     \
168         libmvec         \
169         libndmp         \
170         libnisdb        \
171         libnls          \
172         libnsctl        \
173         libnwam         \
174         libofmt         \
175         libpam          \
176         libpcidb        \
177         libpctx         \
178         libpicl         \
179         libpicltree     \
180         libpkg          \
181         libpool         \
182         libpp           \
183         libproc         \
184         libproject      \
185         libpthread      \
186         libraidcfg      \
187         librcm          \
188         librdc          \
189         libreparse      \
190         libresolv       \
191         libresolv2      \
192         librestart      \
193         librpcsvc       \
```



```
194         librsm          \
195         librstp         \
196         librt           \
197         libsasl         \
198         libscf          \
199         libsched        \
200         libsctp         \
201         libsec          \
202         libsecdb        \
203         libsendfile     \
204         libsff          \
205         libshare        \
206         libshell        \
207         libsip          \
208         libsldap        \
209         libslp          \
210         libsmbfs        \
211         libsmbios       \
212         libsmedia       \
213         libsqlite       \
214         libsrpt         \
215         libstmf         \
216         libstmfproxy    \
217         libsum          \
218         libsun_ima      \
219         libsys          \
220         libsysevent     \
221         libtecla        \
222         libtermcap      \
223         libthread       \
224         libtnf          \
225         libtnfctl       \
226         libtnfprobe     \
227         libtsnet        \
228         libtsol         \
229         libumem         \
230         libunistat      \
231         libuuid         \
232         libuutil        \
233         libvolmgt       \
234         libvrrpadm      \
235         libvscan        \
236         libw            \
237         libwrap         \
238         libxcurses      \
239         libxcurses2     \
240         libxnet         \
241         libzfs          \
242         libzfs_core     \
243         libzfs_jni      \
244         libzonecfg      \
245         libzoneinfo     \
246         libzonestat     \
247         libzpool        \
248         madv            \
249         mpapi           \
250         mpss            \
251         nametoaddr      \
252         ncad_addr       \
253         nsswitch        \
254         pam_modules     \
255         passwdutil      \
256         pkcs11          \
257         policykit       \
258         print           \
259         pylibbe         \
```

```
260         pysolaris       \
261         pyzfs           \
262         raidcfg_plugins \
263         rpcsec_gss      \
264         sasl_plugins    \
265         scsi            \
266         smbsrv          \
267         smhba           \
268         sun_fc          \
269         sun_sas         \
270         udapl           \
271         watchmalloc     \
272         $($(MACH)_SUBDIRS)

274 i386_SUBDIRS=           \
275         libfdisk        \
276         libsaveargs

278 sparc_SUBDIRS=          \
279         efcode          \
280         libds           \
281         libdscp         \
282         libprtdiag      \
283         libprtdiag_psr  \
284         libpri          \
285         librsc          \
286         storage         \
287         libpcp          \
288         libtsalarm      \
289         libv12n

291 #
292 # Create a special version of $(SUBDIRS) with no .WAIT's, for use with the
293 # clean and clobber targets (for more information, see those targets, below).
294 #
295 NOWAIT_SUBDIRS= $(SUBDIRS:.WAIT=)

297 DCSUBDIRS=

299 MSGSUBDIRS=             \
300         abi             \
301         auditd_plugins  \
302         brand           \
303         cfgadm_plugins  \
304         gss_mechs/mech_dh       \
305         gss_mechs/mech_krb5     \
306         krb5            \
307         libast          \
308         libbsm          \
309         libc            \
310         libcfgadm       \
311         libcmd          \
312         libcontract     \
313         libcurses       \
314         libdhcputil     \
315         libdiskmgt      \
316         libdladm        \
317         libdll          \
318         libgrubmgmt     \
319         libgss          \
320         libidmap        \
321         libilb          \
322         libinstzones    \
323         libipadm        \
324         libipmp         \
325         libipsecutil    \
```



```
326         libldap5        \
327         libnsl          \
328         libnwam         \
329         libofmt         \
330         libpam          \
331         libpicl         \
332         libpkg          \
333         libpool         \
334         libpp           \
335         libreparse      \
336         libsasl         \
337         libscf          \
338         libsecdb        \
339         libshare        \
340         libshell        \
341         libsldap        \
342         libslp          \
343         libsmbfs        \
344         libsmedia       \
345         libsum          \
346         libtsol         \
347         libuutil        \
348         libvrrpadm      \
349         libvscan        \
350         libzfs          \
351         libzonecfg      \
352         madv            \
353         mpss            \
354         pam_modules     \
355         pyzfs           \
356         rpcsec_gss      \
357         $($(MACH)_MSGSUBDIRS)

359 sparc_MSGSUBDIRS=       \
360         libprtdiag      \
361         libprtdiag_psr

363 i386_MSGSUBDIRS= libfdisk

365 HDRSUBDIRS=                             \
366         ../cmd/sendmail/libmilter       \
367         auditd_plugins  \
368         fm              \
369         hal             \
370         hbaapi          \
371         libads          \
372         libast          \
373         libbrand        \
374         libbsm          \
375         libc            \
376         libcmd          \
377         libcmdutils     \
378         libcommputil    \
379         libcontract     \
380         libcpc          \
381         libcryptoutil   \
382         libctf          \
383         libcurses       \
384         libdevice       \
385         libdevid        \
386         libdevinfo      \
387         libdhcpagent    \
388         libdhcputil     \
389         libdisasm       \
390         libdiskmgt      \
391         libdladm        \
```

```
392         libdll          \
393         libdlpi         \
394         libdns_sd       \
395         libdscfg        \
396         libdtrace       \
397         libdtrace_jni   \
398         libelfsign      \
399         libeti          \
400         libfakekernel   \
401         libfcoe         \
402         libfru          \
403         libfstyp        \
404         libgen          \
405         libgrubmgmt     \
406         libidmap        \
407         libilb          \
408         libima          \
409         libinetsvc      \
410         libinetutil     \
411         libinstzones    \
412         libipadm        \
413         libipd          \
414         libipmi         \
415         libipmp         \
416         libipp          \
417         libipsecutil    \
418         libiscsit       \
419         libkrb5         \
420         libkstat        \
421         libkvm          \
422         libmail         \
423         libmapid        \
424         libmd           \
425         libmlrpc        \
426         libmtmalloc     \
427         libndmp         \
428         libnsctl        \
429         libnsl          \
430         libnvpair       \
431         libnwam         \
432         libofmt         \
433         libpam          \
434         libpcidb        \
435         libpctx         \
436         libpicl         \
437         libpicltree     \
438         libpool         \
439         libpp           \
440         libproc         \
441         libraidcfg      \
442         librcm          \
443         librdc          \
444         libreparse      \
445         librestart      \
446         librpcsvc       \
447         librsm          \
448         librstp         \
449         libsasl         \
450         libscf          \
451         libsec          \
452         libsff          \
453         libshare        \
454         libshell        \
455         libsip          \
456         libslp          \
457         libsmbfs        \
```



```
458         libsmbios       \
459         libsmedia       \
460         libsocket       \
461         libsqlite       \
462         libsrpt         \
463         libstmf         \
464         libstmfproxy    \
465         libsum          \
466         libsun_ima      \
467         libsysevent     \
468         libtecla        \
469         libtermcap      \
470         libtnf          \
471         libtnfctl       \
472         libtnfprobe     \
473         libtsnet        \
474         libtsol         \
475         libumem         \
476         libunistat      \
477         libuutil        \
478         libvolmgt       \
479         libvrrpadm      \
480         libvscan        \
481         libwrap         \
482         libxcurses2     \
483         libzfs          \
484         libzfs_core     \
485         libzfs_jni      \
486         libzoneinfo     \
487         libzonestat     \
488         mpapi           \
489         passwdutil      \
490         pkcs11          \
491         policykit       \
492         scsi            \
493         smbsrv          \
494         smhba           \
495         udapl           \
496         $($(MACH)_HDRSUBDIRS)

498 i386_HDRSUBDIRS=       \
499         libfdisk        \
500         libsaveargs

502 sparc_HDRSUBDIRS=       \
503         libds           \
504         libdscp         \
505         libpri          \
506         libv12n         \
507         storage

509 all :=          TARGET= all
510 check :=        TARGET= check
511 clean :=        TARGET= clean
512 clobber :=      TARGET= clobber
513 install :=      TARGET= install
514 install_h :=    TARGET= install_h
515 lint :=         TARGET= lint
516 _dc :=          TARGET= _dc
517 _msg :=         TARGET= _msg

519 .KEEP_STATE:

521 #
522 # For the all and install targets, we clearly must respect library
523 # dependencies so that the libraries link correctly.  However, for
```

```
524 # the remaining targets (check, clean, clobber, install_h, lint, _dc
525 # and _msg), libraries do not have any dependencies on one another
526 # and thus respecting dependencies just slows down the build.
527 # As such, for these rules, we use pattern replacement to explicitly
528 # avoid triggering the dependency information.  Note that for clean,
529 # clobber and lint, we must use $(NOWAIT_SUBDIRS) rather than
530 # $(SUBDIRS), to prevent '.WAIT' from expanding to '.WAIT-nodepend'.
531 #

533 all:                    $(SUBDIRS)

535 install:                $(SUBDIRS) .WAIT install_extra

537 # extra libraries kept in other source areas
538 install_extra:
539         @cd ../cmd/sgs; pwd; $(MAKE) install_lib
540         @pwd

542 clean clobber lint:     $(NOWAIT_SUBDIRS:%=%-nodepend)

544 install_h check:        $(HDRSUBDIRS:%=%-nodepend)

546 _msg:                   $(MSGSUBDIRS:%=%-nodepend) .WAIT _dc

548 _dc:                    $(DCSUBDIRS:%=%-nodepend)

550 # Library interdependencies are called out explicitly here.
551 ../cmd/sgs/libconv: crt
552 ../cmd/sgs/libdl: crt
553 libc:           ../cmd/sgs/libconv ../cmd/sgs/libdl

555 libm:           libc
556 libmd:          libc
557 libmp:          libc
558 libnsl:         libc libmd libmp
559 libnvpair:      libc libnsl
560 libsocket:      libc libnsl

562 basedeps:       libc libm libmd libmp libnsl libnvpair libsocket
563 basedeps :=     TARGET=install

565 # The following dependencies are currently required by libdbus-1 and should
566 # be used for anything linking against it.
567 dbusdeps:       libsecdb libtsol libinetutil libscf libuutil libgen libsmbios

569 # The following dependencies don't need to be listed below as they are built
570 # before everything else:
571 # libc libm libmd libmp libnsl libnvpair libsocket
572 abi:            libctf libmapmalloc libproc
573 auditd_plugins: libbsm libsecdb libgss libmtmalloc
574 brand:          libzonecfg libmapmalloc
575 cfgadm_plugins: libdevice libdevinfo libhotplug librcm hbaapi libkstat libscf
576 fm:             libexacct libipmi libzfs scsi libdevinfo libdevid libcfgadm \
577                 libcontract libsysevent ../cmd/sgs/libelf libdladm libsff
578 $(SPARC_BLD)fm: libpri
579 gss_mechs/mech_dh: libgss
580 gss_mechs/mech_dummy: libgss
581 gss_mechs/mech_krb5: libgss libresolv2 pkcs11 libkstat
582 gss_mechs/mech_spnego: gss_mechs/mech_krb5
583 hal:            dbusdeps
584 krb5:           gss_mechs/mech_krb5 libtecla libldap5
585 libadt_jni:     libbsm
586 libadutils:     libldap5 libresolv2
587 libbe:          libzfs libinstzones libuuid libgen libdevinfo libefi libficl
588 libbsm:         libinetutil libscf libsecdb libtsol
589 libcfgadm:      libdevinfo
```



```
590 libcmd:          libsum libast
591 libcmdutils:    libavl
592 libcpc:         libpctx
593 libcrypt:       libgen
594 libdevid:       libdevinfo
595 libdevinfo:     libsec libgen
596 libdhcpagent:   libdhcputil libuuid libdlpi libcontract
597 libdhcputil:    libgen libinetutil libdlpi
598 libdiskmgt:     libdevid libdevinfo libadm libefi libkstat libsysevent
599 $(INTEL_BLD)libdiskmgt: libfdisk
600 libdladm:       libdevinfo libinetutil libscf librcm libexacct libkstat \
601                 libpool
602 libdll:         libast
603 libdlpi:        libinetutil libdladm
604 libds:          libsysevent
605 libdscfg:       libnsctl libunistat libadm
606 libdtrace:      libproc libgen libctf libmapmalloc
607 libdtrace_jni:  libuutil libdtrace
608 libefi:         libuuid
609 libelfsign:     libcryptoutil libkmf ../cmd/sgs/libelf
610 libeti:         libcurses
611 libexacct/demo: libexacct libproject
612 libfakekernel:  libumem libcryptoutil
613 libfcoe:        libdladm
614 libficl:        libuuid libumem
615 libfru:         libfruutils
616 libfsmgt:       libkstat
617 libgrubmgmt:    libdevinfo libzfs libfstyp libefi
618 $(INTEL_BLD)libgrubmgmt: libfdisk
619 libidmap:       libavl libuutil
620 libinetsvc:     libscf
621 libinstzones:   libzonecfg libcontract
622 libipadm:       libinetutil libdlpi libdhcpagent libdladm libsecdb libdhcputil
623 libipmp:        libinetutil
624 libipsecutil:   libtecla libtsol
625 libiscsit:      libstmf libuuid
626 libkmf:         libcryptoutil pkcs11
627 libkvm:         ../cmd/sgs/libelf
628 libldap5:       libsasl
629 libmapid:       libresolv2 libscf
630 libmlrpc:       libsmbfs libuuid
631 libndmp:        libscf
632 libnisdb:       libldap5
633 libnwam:        libscf libbsm libdladm libipadm
634 libpcp:         libumem libdevinfo
635 libpctx:        libproc
636 libpkg:         libscf libadm
637 libpool:        libscf libexacct
638 libpp:          libast
639 libproc:        ../cmd/sgs/librtld_db ../cmd/sgs/libelf libctf
640 $(INTEL_BLD)libproc: libsaveargs
641 libproject:     libpool libproc libsecdb
642 libprtdiag:     libkstat
643 libprtdiag_psr: libprtdiag
644 libraidcfg:     libdevinfo
645 librdc:         libnsctl libunistat libdscfg
646 librestart:     libuutil libscf libpool libproject libsecdb libsysevent
647 libsasl:        libgss pkcs11
648 libsaveargs:    libdisasm
649 libscf:         libuutil libgen libsmbios
650 libsec:         libavl libidmap
651 libsff:         libnvpair
652 libshare:       libscf libzfs libuuid libfsmgt libsecdb libumem libsmbfs
653 libshell:       libast libcmd libdll libsecdb
654 libsip:         libmd5
655 libsldap:       libldap5 libscf
```

```
libsmbfs:       libkrb5 libsec libidmap pkcs11
libsmbios:      libdevinfo
libsrpt:        libstmf
libstmf:        libscf
libstmfproxy:   libstmf libpthread
libsum:         libast
libsun_ima:     libdevinfo libsysevent
libsysevent:    libsecdb
libtecla:       libcurses
libtermcap:     libcurses
libtnfctl:      ../cmd/sgs/libelf
libtsalarm:     libpcp
libtsnet:       libtsol libsecdb
libtsol:        libsecdb
libuuid:        libdlpi
libv12n:        libds libuuid
libvolmgt:      libadm
libvrrpadm:     libdladm libscf
libvscan:       libscf libsecdb
libzfs:         libdevid libgen libuutil libadm libavl libefi libidmap \
                libumem libtsol libzfs_core libcmdutils
libzfs_jni:     libdiskmgt libzfs
libzonecfg:     libuuid libsysevent libsec libbrand libpool libscf libproc \
                libuutil libbsm libsecdb
libzonestat:    libcmdutils libumem
libzpool:       libavl libumem libcmdutils libsysevent libfakekernel
madv:           libgen
mpapi:          libpthread libdevinfo libsysevent
mpss:           libgen
nsswitch:       libadutils libidmap libdns_sd libscf libldap5 libsldap
pam_modules:    libproject passwdutil smbsrv libtsnet libpam libbsm libsecdb
passwdutil:     libsldap
pkcs11:         libcryptoutil libgen libuuid
policykit:      dbusdeps
print:          libldap5 libmd5 libsendfile
pylibbe:        libbe libzfs
pysolaris:      libsec libidmap
pyzfs:          libzfs
raidcfg_plugins: libraidcfg librcm libcfgadm libpicl libpicltree
rpcsec_gss:     libgss
sasl_plugins:   pkcs11 libgss libsasl
scsi:           libfru libumem libdevid libdevinfo
smbsrv:         libxnet libpthread librt libshare libidmap pkcs11 libsqlite \
                libcryptoutil libreparse libcmdutils libresolv2 libsmbfs \
                libuuid libfakekernel libads libgss libldap5 krb5 libmlrpc
                libuuid libfakekernel libads libgss libldap5 krb5
storage:        libdevice libdevinfo libdevid
sun_fc:         libdevinfo libsysevent
sun_sas:        libdevinfo libsysevent libkstat libdevid
udapl:          libdevinfo libdladm

#
# The reason this rule checks for the existence of the
# Makefile is that some of the directories do not exist
# in certain situations (e.g., exportable source builds,
# OpenSolaris).
#
$(SUBDIRS): FRC
        @if [ -f $@/Makefile  ]; then \
                cd $@; pwd; $(MAKE) $(TARGET); \
        else \
                true; \
        fi

$(SUBDIRS:%=%-nodepend):
        @if [ -f $(@:%-nodepend=%)/Makefile  ]; then \
```



```
                cd $(@:%-nodepend=%); pwd; $(MAKE) $(TARGET); \
        else \
                true; \
        fi

FRC:
```

```
**********************************************************
    1616 Sun Mar 18 01:12:57 2018
new/usr/src/lib/libmlrpc/Makefile
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
# ident "%Z%%M% %I%     %E% SMI"
#

include $(SRC)/lib/Makefile.lib
HDRS=   libmlrpc.h

HDRS=           libmlrpc.h ndr.h ndrtypes.ndl rpcpdu.ndl
HDRDIR=         common

ROOTHDRDIR=     $(ROOT)/usr/include/libmlrpc
ROOTHDRS= $(HDRS:%=$(ROOTHDRDIR)/%)

# ISA targets
SUBDIRS = $(MACH)
$(BUILD64)SUBDIRS += $(MACH64)

all :=          TARGET = all
install :=      TARGET = install
clean :=        TARGET = clean
clobber :=      TARGET = clobber
lint :=         TARGET = lint

.KEEP_STATE:

all install clean clobber lint: $(SUBDIRS)

install_h: $(ROOTHDRDIR) $(ROOTHDRS)

check: $(CHECKHDRS)

$(ROOTHDRDIR)/%: %
        $(INS.file)

$(ROOTHDRDIR):
        $(INS.dir)
```



```
$(SUBDIRS): FRC
        @cd $@; pwd; VERSION='$(VERSION)' $(MAKE) $(TARGET)

FRC:

include $(SRC)/lib/Makefile.targ
include ../Makefile.smbsrv
```

```
**********************************************************
    1727 Sun Mar 18 01:12:57 2018
new/usr/src/lib/libmlrpc/Makefile.com
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
#

LIBRARY =       libmlrpc.a
VERS =          .2
VERS =          .1

OBJS_COMMON =                   \
        mlrpc_clh.o             \
        ndr_client.o            \
        ndr_heap.o              \
        ndr_marshal.o           \
        ndr_ops.o               \
        ndr_process.o           \
        ndr_server.o            \
        ndr_svc.o               \
        ndr_wchar.o
        ndr_svc.o

NDLLIST = rpcpdu

OBJECTS=        $(OBJS_COMMON) $(NDLLIST:%=%_ndr.o)
CLEANFILES += $(NDLLIST:%=%_ndr.c)
OBJECTS=        $(OBJS_COMMON) $(OBJS_SHARED) $(NDLLIST:%=%_ndr.o)

include ../../../Makefile.lib
include ../../Makefile.lib

LIBS=           $(DYNLIB) $(LINTLIB)
INCS += -I$(SRC)/common/smbsrv

LDLIBS +=       -lsmbfs -luuid -lc
LDLIBS +=       $(MACH_LDLIBS)
LDLIBS +=       -lsmb -luuid -lc

SRCDIR=         ../common
```



```
SRCS=   $(OBJS_COMMON:%.o=$(SRCDIR)/%.c)
$(LINTLIB) := SRCS = $(SRCDIR)/$(LINTSRC)

NDLDIR =        $(SRCDIR)

CFLAGS +=       $(CCVERBOSE)
INCS = -I. -I$(SRCDIR)
CPPFLAGS += $(INCS) -D_REENTRANT

all:    $(LIBS)
SRCS=   $(OBJS_COMMON:%.o=$(SRCDIR)/%.c)                \
        $(OBJS_SHARED:%.o=$(SRC)/common/smbsrv/%.c)

lint:   lintcheck

include ../../Makefile.targ

objs/%_ndr.o pics/%_ndr.o : %_ndr.c

%_ndr.c : $(NDLDIR)/%.ndl
        $(NDRGEN) -Y $(ANSI_CPP) $(CPPFLAGS) $<

.KEEP_STATE:
include ../../../Makefile.targ
```

```
**********************************************************
    1063 Sun Mar 18 01:12:57 2018
new/usr/src/lib/libmlrpc/amd64/Makefile
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
   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 2007 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #
  25 # ident "%Z%%M% %I%     %E% SMI"
  26 #

  26 MACH_LDLIBS +=  -L$(ROOT)/usr/lib/smbsrv/$(MACH64)

  28 include ../Makefile.com
  29 include ../../Makefile.lib.64
  31 include ../../../Makefile.lib.64

  33 DYNFLAGS +=     -R/usr/lib/smbsrv/$(MACH64)

  31 install: all $(ROOTLIBS64) $(ROOTLINKS64) $(ROOTLINT64)
```

```
**********************************************************
   19102 Sun Mar 18 01:12:58 2018
new/usr/src/lib/libmlrpc/common/libmlrpc.h
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
   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) 2007, 2010, Oracle and/or its affiliates. All rights reserved.
  23  * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
  24  */

  26 #ifndef _LIBMLRPC_H
  27 #define _LIBMLRPC_H

  29 #include <sys/types.h>
  30 #include <sys/uio.h>
  31 #include <smbsrv/wintypes.h>
  32 #include <smbsrv/ndr.h>
  33 #include <smbsrv/smb_sid.h>
  34 #include <smbsrv/smb_xdr.h>

  32 #include <smb/wintypes.h>
  33 #include <libmlrpc/ndr.h>

  35 #ifdef  __cplusplus
  36 extern "C" {
  37 #endif

  39 /*
  40  * An MSRPC compatible implementation of OSF DCE RPC.  DCE RPC is derived
  41  * from the Apollo Network Computing Architecture (NCA) RPC implementation.
  42  *
  43  * CAE Specification (1997)
  44  * DCE 1.1: Remote Procedure Call
  45  * Document Number: C706
  46  * The Open Group
  47  * ogspecs@opengroup.org
  48  *
  49  * This implementation is based on the DCE Remote Procedure Call spec with
  50  * enhancements to support Unicode strings.  The diagram below shows the
  51  * DCE RPC layers compared against ONC SUN RPC.
  52  *
  53  *      NDR RPC Layers          Sun RPC Layers          Remark
  54  *      +---------------+       +---------------+       +---------------+
  55  *      +---------------+       +---------------+
  56  *      | Application   |       | Application   |       The application
  57  *      +---------------+       +---------------+
```



```
  58  *      | Hand coded    |       | RPCGEN gen'd  |       Where the real
  59  *      | client/server |       | client/server |       work happens
  60  *      | srvsvc.ndl    |       | *_svc.c *_clnt|
  61  *      | srvsvc.c      |       |               |
  62  *      +---------------+       +---------------+
  63  *      | RPC Library   |       | RPC Library   |       Calls/Return
  64  *      | ndr_*.c       |       |               |       Binding/PMAP
  65  *      +---------------+       +---------------+
  66  *      | RPC Protocol  |       | RPC Protocol  |       Headers, Auth,
  67  *      | rpcpdu.ndl    |       |               |
  68  *      +---------------+       +---------------+
  69  *      | IDL gen'd     |       | RPCGEN gen'd  |       Aggregate
  70  *      | NDR stubs     |       | XDR stubs     |       Composition
  71  *      | *__ndr.c      |       | *_xdr.c       |
  72  *      +---------------+       +---------------+
  73  *      | NDR Represen  |       | XDR Represen  |       Byte order, padding
  74  *      +---------------+       +---------------+
  75  *      | Packet Heaps  |       | Network Conn  |       DCERPC does not talk
  76  *      | ndo_*.c       |       | clnt_{tcp,udp}|       directly to network.
  77  *      +---------------+       +---------------+
  78  *
  79  * There are two major differences between the DCE RPC and ONC RPC:
  80  *
  81  * 1. NDR RPC only generates or processes packets from buffers.  Other
  82  *    layers must take care of packet transmission and reception.
  83  *    The packet heaps are managed through a simple interface provided
  84  *    by the Network Data Representation (NDR) module called ndr_stream_t.
  85  *    ndo_*.c modules implement the different flavors (operations) of
  86  *    packet heaps.
  87  *
  88  *    ONC RPC communicates directly with the network.  You have to do
  89  *    something special for the RPC packet to be placed in a buffer
  90  *    rather than sent to the wire.
  91  *
  92  * 2. NDR RPC uses application provided heaps to support operations.
  93  *    A heap is a single, monolithic chunk of memory that NDR RPC manages
  94  *    as it allocates.  When the operation and its result are done, the
  95  *    heap is disposed of as a single item.  The transaction, which
  96  *    is the anchor of most operations, contains the necessary book-
  97  *    keeping for the heap.
  98  *
  99  *    ONC RPC uses malloc() liberally throughout its run-time system.
 100  *    To free results, ONC RPC supports an XDR_FREE operation that
 101  *    traverses data structures freeing memory as it goes, whether
 102  *    it was malloc'd or not.
 103  */

 105 /*
 106  * Dispatch Return Code (DRC)
 107  *
 108  *      0x8000  15:01   Set to indicate a fault, clear indicates status
 109  *      0x7F00  08:07   Status/Fault specific
 110  *      0x00FF  00:08   PTYPE_... of PDU, 0xFF for header
 111  */
 112 #define NDR_DRC_OK                              0x0000
 113 #define NDR_DRC_MASK_FAULT                      0x8000
 114 #define NDR_DRC_MASK_SPECIFIER                  0xFF00
 115 #define NDR_DRC_MASK_PTYPE                      0x00FF

 117 /* Fake PTYPE DRC discriminators */
 118 #define NDR_DRC_PTYPE_RPCHDR(DRC)               ((DRC) | 0x00FF)
 119 #define NDR_DRC_PTYPE_API(DRC)                  ((DRC) | 0x00AA)

 121 /* DRC Recognizers */
 122 #define NDR_DRC_IS_OK(DRC)      (((DRC) & NDR_DRC_MASK_SPECIFIER) == 0)
 123 #define NDR_DRC_IS_FAULT(DRC)   (((DRC) & NDR_DRC_MASK_FAULT) != 0)
```

```
125 /*
126  * (Un)Marshalling category specifiers
127  */
128 #define NDR_DRC_FAULT_MODE_MISMATCH             0x8100
129 #define NDR_DRC_RECEIVED                        0x0200
130 #define NDR_DRC_FAULT_RECEIVED_RUNT             0x8300
131 #define NDR_DRC_FAULT_RECEIVED_MALFORMED        0x8400
132 #define NDR_DRC_DECODED                         0x0500
133 #define NDR_DRC_FAULT_DECODE_FAILED             0x8600
134 #define NDR_DRC_ENCODED                         0x0700
135 #define NDR_DRC_FAULT_ENCODE_FAILED             0x8800
136 #define NDR_DRC_FAULT_ENCODE_TOO_BIG            0x8900
137 #define NDR_DRC_SENT                            0x0A00
138 #define NDR_DRC_FAULT_SEND_FAILED               0x8B00

140 /*
141  * Resource category specifier
142  */
143 #define NDR_DRC_FAULT_RESOURCE_1                0x9100
144 #define NDR_DRC_FAULT_RESOURCE_2                0x9200

146 /*
147  * Parameters. Usually #define'd with useful alias
148  */
149 #define NDR_DRC_FAULT_PARAM_0_INVALID           0xC000
150 #define NDR_DRC_FAULT_PARAM_0_UNIMPLEMENTED     0xD000
151 #define NDR_DRC_FAULT_PARAM_1_INVALID           0xC100
152 #define NDR_DRC_FAULT_PARAM_1_UNIMPLEMENTED     0xD100
153 #define NDR_DRC_FAULT_PARAM_2_INVALID           0xC200
154 #define NDR_DRC_FAULT_PARAM_2_UNIMPLEMENTED     0xD200
155 #define NDR_DRC_FAULT_PARAM_3_INVALID           0xC300
156 #define NDR_DRC_FAULT_PARAM_3_UNIMPLEMENTED     0xD300

158 #define NDR_DRC_FAULT_OUT_OF_MEMORY             0xF000

160 /* RPCHDR */
161 #define NDR_DRC_FAULT_RPCHDR_MODE_MISMATCH      0x81FF
162 #define NDR_DRC_FAULT_RPCHDR_RECEIVED_RUNT      0x83FF
163 #define NDR_DRC_FAULT_RPCHDR_DECODE_FAILED      0x86FF
164 #define NDR_DRC_FAULT_RPCHDR_PTYPE_INVALID      0xC0FF  /* PARAM_0_INVALID */
165 #define NDR_DRC_FAULT_RPCHDR_PTYPE_UNIMPLEMENTED 0xD0FF /* PARAM_0_UNIMP */

167 /* Request */
168 #define NDR_DRC_FAULT_REQUEST_PCONT_INVALID     0xC000  /* PARAM_0_INVALID */
169 #define NDR_DRC_FAULT_REQUEST_OPNUM_INVALID     0xC100  /* PARAM_1_INVALID */

171 /* Bind */
172 #define NDR_DRC_BINDING_MADE                    0x000B  /* OK */
173 #define NDR_DRC_FAULT_BIND_PCONT_BUSY           0xC00B  /* PARAM_0_INVALID */
174 #define NDR_DRC_FAULT_BIND_UNKNOWN_SERVICE      0xC10B  /* PARAM_1_INVALID */
175 #define NDR_DRC_FAULT_BIND_NO_SLOTS             0x910B  /* RESOURCE_1 */

177 /* API */
178 #define NDR_DRC_FAULT_API_SERVICE_INVALID       0xC0AA  /* PARAM_0_INVALID */
179 #define NDR_DRC_FAULT_API_BIND_NO_SLOTS         0x91AA  /* RESOURCE_1 */
180 #define NDR_DRC_FAULT_API_OPNUM_INVALID         0xC1AA  /* PARAM_1_INVALID */

182 struct ndr_xa;
183 struct ndr_client;

185 typedef struct ndr_stub_table {
186         int             (*func)(void *, struct ndr_xa *);
187         unsigned short  opnum;
188 } ndr_stub_table_t;
```

*_____unchanged_portion_omitted_*


```
235 #define NDR_BIND_SIDE_CLIENT    1
236 #define NDR_BIND_SIDE_SERVER    2

238 #define NDR_BINDING_TO_SPECIFIC(BINDING, TYPE) \
239         ((TYPE *) (BINDING)->instance_specific)

241 /*
242  * The binding list space must be provided by the application library
243  * for use by the underlying RPC library.  We need at least two binding
244  * slots per connection.
245  */
246 #define NDR_N_BINDING_POOL      2

248 typedef struct ndr_pipe {
249         void                    *np_listener;
250         const char              *np_endpoint;
251         struct smb_netuserinfo  *np_user;
252         smb_netuserinfo_t       *np_user;
252         int                     (*np_send)(struct ndr_pipe *, void *, size_t);
253         int                     (*np_recv)(struct ndr_pipe *, void *, size_t);
254         int                     np_fid;
255         uint16_t                np_max_xmit_frag;
256         uint16_t                np_max_recv_frag;
257         ndr_binding_t           *np_binding;
258         ndr_binding_t           np_binding_pool[NDR_N_BINDING_POOL];
259 } ndr_pipe_t;
```

*_____unchanged_portion_omitted_*

```
402 ndr_heap_t *ndr_heap_create(void);
403 void ndr_heap_destroy(ndr_heap_t *);
404 void *ndr_heap_dupmem(ndr_heap_t *, const void *, size_t);
405 void *ndr_heap_malloc(ndr_heap_t *, unsigned);
406 void *ndr_heap_strdup(ndr_heap_t *, const char *);
407 int ndr_heap_mstring(ndr_heap_t *, const char *, ndr_mstring_t *);
408 void ndr_heap_mkvcs(ndr_heap_t *, char *, ndr_vcstr_t *);
409 void ndr_heap_mkvcb(ndr_heap_t *, uint8_t *, uint32_t, ndr_vcbuf_t *);
410 smb_sid_t *ndr_heap_siddup(ndr_heap_t *, smb_sid_t *);
410 int ndr_heap_used(ndr_heap_t *);
411 int ndr_heap_avail(ndr_heap_t *);

413 #define NDR_MALLOC(XA, SZ)      ndr_heap_malloc((XA)->heap, SZ)
414 #define NDR_NEW(XA, T)          ndr_heap_malloc((XA)->heap, sizeof (T))
415 #define NDR_NEWN(XA, T, N)      ndr_heap_malloc((XA)->heap, sizeof (T)*(N))
416 #define NDR_STRDUP(XA, S)       ndr_heap_strdup((XA)->heap, (S))
417 #define NDR_MSTRING(XA, S, OUT) ndr_heap_mstring((XA)->heap, (S), (OUT))
418 #define NDR_SIDDUP(XA, S)       ndr_heap_dupmem((XA)->heap, (S), smb_sid_len(S))
419 #define NDR_SIDDUP(XA, S)       ndr_heap_siddup((XA)->heap, (S))

420 typedef struct ndr_xa {
421         unsigned short          ptype;          /* high bits special */
422         unsigned short          opnum;
423         ndr_stream_t            recv_nds;
424         ndr_hdr_t               recv_hdr;
425         ndr_stream_t            send_nds;
426         ndr_hdr_t               send_hdr;
427         ndr_binding_t           *binding;       /* what we're using */
428         ndr_binding_t           *binding_list;  /* from connection */
429         ndr_heap_t              *heap;
430         ndr_pipe_t              *pipe;
431 } ndr_xa_t;
```

*_____unchanged_portion_omitted_*

```
484 /* ndr_ops.c */
485 int nds_initialize(ndr_stream_t *, unsigned, int, ndr_heap_t *);
486 void nds_destruct(ndr_stream_t *);
```

```
void nds_show_state(ndr_stream_t *);

/* ndr_client.c */
int ndr_clnt_bind(ndr_client_t *, ndr_service_t *, ndr_binding_t **);
int ndr_clnt_bind(ndr_client_t *, const char *, ndr_binding_t **);
int ndr_clnt_call(ndr_binding_t *, int, void *);
void ndr_clnt_free_heap(ndr_client_t *);

/* ndr_marshal.c */
ndr_buf_t *ndr_buf_init(ndr_typeinfo_t *);
void ndr_buf_fini(ndr_buf_t *);
int ndr_buf_decode(ndr_buf_t *, unsigned, unsigned, const char *data, size_t,
    void *);
int ndr_decode_call(ndr_xa_t *, void *);
int ndr_encode_return(ndr_xa_t *, void *);
int ndr_encode_call(ndr_xa_t *, void *);
int ndr_decode_return(ndr_xa_t *, void *);
int ndr_decode_pdu_hdr(ndr_xa_t *);
int ndr_encode_pdu_hdr(ndr_xa_t *);
void ndr_decode_frag_hdr(ndr_stream_t *, ndr_common_header_t *);
void ndr_remove_frag_hdr(ndr_stream_t *);
void ndr_show_hdr(ndr_common_header_t *);
unsigned ndr_bind_ack_hdr_size(ndr_xa_t *);
unsigned ndr_alter_context_rsp_hdr_size(void);

/* ndr_server.c */
void ndr_pipe_worker(ndr_pipe_t *);

int ndr_generic_call_stub(ndr_xa_t *);

boolean_t ndr_is_admin(ndr_xa_t *);
boolean_t ndr_is_poweruser(ndr_xa_t *);
int32_t ndr_native_os(ndr_xa_t *);

/* ndr_svc.c */
ndr_stub_table_t *ndr_svc_find_stub(ndr_service_t *, int);
ndr_service_t *ndr_svc_lookup_name(const char *);
ndr_service_t *ndr_svc_lookup_uuid(ndr_uuid_t *, int, ndr_uuid_t *, int);
int ndr_svc_register(ndr_service_t *);
void ndr_svc_unregister(ndr_service_t *);
void ndr_svc_binding_pool_init(ndr_binding_t **, ndr_binding_t pool[], int);
ndr_binding_t *ndr_svc_find_binding(ndr_xa_t *, ndr_p_context_id_t);
ndr_binding_t *ndr_svc_new_binding(ndr_xa_t *);

int ndr_uuid_parse(char *, ndr_uuid_t *);
void ndr_uuid_unparse(ndr_uuid_t *, char *);

ndr_hdid_t *ndr_hdalloc(const ndr_xa_t *, const void *);
void ndr_hdfree(const ndr_xa_t *, const ndr_hdid_t *);
ndr_handle_t *ndr_hdlookup(const ndr_xa_t *, const ndr_hdid_t *);
void ndr_hdclose(ndr_pipe_t *);

ssize_t ndr_uiomove(caddr_t, size_t, enum uio_rw, struct uio *);

/*
 * An ndr_client_t is created while binding a client connection to hold
 * the context for calls made using that connection.
 *
 * Handles are RPC call specific and we use an inheritance mechanism to
 * ensure that each handle has a pointer to the client_t.  When the top
 * level (bind) handle is released, we close the connection.
 *
 * There are some places in libmlsvc where the code assumes that the
 * handle member is first in this struct.  careful
 */
typedef struct mlrpc_handle {
```



```
        ndr_hdid_t      handle;         /* keep first */
        ndr_client_t    *clnt;
} mlrpc_handle_t;

int mlrpc_clh_create(mlrpc_handle_t *, void *);
uint32_t mlrpc_clh_bind(mlrpc_handle_t *, ndr_service_t *);
void mlrpc_clh_unbind(mlrpc_handle_t *);
void *mlrpc_clh_free(mlrpc_handle_t *);

int ndr_rpc_call(mlrpc_handle_t *, int, void *);
int ndr_rpc_get_ssnkey(mlrpc_handle_t *, unsigned char *, size_t);
void *ndr_rpc_malloc(mlrpc_handle_t *, size_t);
ndr_heap_t *ndr_rpc_get_heap(mlrpc_handle_t *);
void ndr_rpc_release(mlrpc_handle_t *);
void ndr_rpc_set_nonull(mlrpc_handle_t *);

boolean_t ndr_is_null_handle(mlrpc_handle_t *);
boolean_t ndr_is_bind_handle(mlrpc_handle_t *);
void ndr_inherit_handle(mlrpc_handle_t *, mlrpc_handle_t *);

#ifdef  __cplusplus
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    1045 Sun Mar 18 01:12:58 2018
new/usr/src/lib/libmlrpc/common/llib-lmlrpc
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
   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 2007 Sun Microsystems, Inc.  All rights reserved.
  23  * Use is subject to license terms.
  24  *
  25  * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
  26  */

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

  28 /*LINTLIBRARY*/
  29 /*PROTOLIB1*/

  31 #include <libmlrpc.h>
  31 #include <smbsrv/libmlrpc.h>
```

```
**********************************************************
    2494 Sun Mar 18 01:12:58 2018
new/usr/src/lib/libmlrpc/common/mapfile-vers
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2007, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
#
#


#
# MAPFILE HEADER START
#
# WARNING:  STOP NOW.  DO NOT MODIFY THIS FILE.
# Object versioning must comply with the rules detailed in
#
#       usr/src/lib/README.mapfiles
#
# You should not be making modifications here until you've read the most current
# copy of that file. If you need help, contact a gatekeeper for guidance.
#
# MAPFILE HEADER END
#

$mapfile_version 2

SYMBOL_VERSION SUNWprivate {
    global:
        mlrpc_clh_bind;
        mlrpc_clh_create;
        mlrpc_clh_free;
        mlrpc_clh_unbind;

        # Allow debug/test programs to provide these.
        ndo_printf              { FLAGS = NODIRECT };
        ndo_trace               { FLAGS = NODIRECT };

        ndr_buf_decode;
        ndr_buf_fini;
        ndr_buf_init;
        ndr_clnt_bind;
        ndr_clnt_call;
        ndr_clnt_free_heap;
        ndr_generic_call_stub;
        ndr_heap_avail;
```



```
        ndr_heap_create;
        ndr_heap_destroy;
        ndr_heap_dupmem;
        ndr_heap_malloc;
        ndr_heap_mkvcb;
        ndr_heap_mkvcs;
        ndr_heap_mstring;
        ndr_heap_siddup;
        ndr_heap_strdup;
        ndr_heap_used;
        ndr_hdalloc;
        ndr_hdclose;
        ndr_hdfree;
        ndr_hdlookup;
        ndr_inherit_handle;
        ndr_inner;
        ndr_is_bind_handle;
        ndr_is_null_handle;
        ndr_is_admin;
        ndr_is_poweruser;
        ndr_mbstowcs;
        ndr_mbtowc;
        ndr_native_os;
        ndr_params;
        ndr_pipe_worker;
        ndr_rpc_call;
        ndr_rpc_get_heap;
        ndr_rpc_get_ssnkey;
        ndr_rpc_malloc;
        ndr_rpc_release;
        ndr_rpc_set_nonull;
        ndr_svc_binding_pool_init;
        ndr_svc_lookup_name;
        ndr_svc_register;
        ndr_topmost;
        ndr_uuid_parse;
        ndr_uuid_unparse;

        nds_destruct;
        nds_initialize;

        ndt__char;
        ndt_s_char;
        ndt__uchar;
        ndt_s_uchar;
        ndt__wchar;
        ndt_s_wchar;
        ndt__short;
        ndt_s_short;
        ndt__ushort;
        ndt_s_ushort;
        ndt__long;
        ndt_s_long;
        ndt__uchar;
        ndt__ulong;
        ndt_s_ulong;

        ndt__ushort;
    local:
        *;
};
```

_____unchanged_portion_omitted_

```
**********************************************************
   13545 Sun Mar 18 01:12:58 2018
new/usr/src/lib/libmlrpc/common/mlrpc_clh.c
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

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

/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 */

/*
 * ML-RPC Client handle interface and support functions.
 */

#include <sys/types.h>
#include <sys/fcntl.h>
#include <sys/poll.h>

#include <errno.h>
#include <strings.h>
#include <unistd.h>

#include <netsmb/smbfs_api.h>
#include <smb/ntstatus.h>
#include <libmlrpc.h>

#include <assert.h>

static int ndr_xa_init(ndr_client_t *, ndr_xa_t *);
static int ndr_xa_exchange(ndr_client_t *, ndr_xa_t *);
static int ndr_xa_read(ndr_client_t *, ndr_xa_t *);
static void ndr_xa_preserve(ndr_client_t *, ndr_xa_t *);
static void ndr_xa_destruct(ndr_client_t *, ndr_xa_t *);
static void ndr_xa_release(ndr_client_t *);

/* See notes in mlrpc_clh_bind */
int rpc_pipe_open_retries = 10;

/*
 * Create an RPC client binding handle using the given smb_ctx.
 * That context must already have a session and tree connected.
 *
 * Returns zero or an errno value.
```



```
 */
int
mlrpc_clh_create(mlrpc_handle_t *handle, void *ctx)
{
        ndr_client_t    *clnt = NULL;

        if (ctx == NULL)
                return (EINVAL);

        /*
         * Allocate...
         */
        if ((clnt = malloc(sizeof (*clnt))) == NULL)
                return (ENOMEM);
        bzero(clnt, sizeof (*clnt));

        clnt->xa_fd = -1;

        /*
         * Setup the transport functions.
         * Always a named pipe (for now).
         */
        clnt->xa_private = ctx;
        clnt->xa_init = ndr_xa_init;
        clnt->xa_exchange = ndr_xa_exchange;
        clnt->xa_read = ndr_xa_read;
        clnt->xa_preserve = ndr_xa_preserve;
        clnt->xa_destruct = ndr_xa_destruct;
        clnt->xa_release = ndr_xa_release;

        /* See _is_bind_handle */
        clnt->handle = &handle->handle;

        ndr_svc_binding_pool_init(&clnt->binding_list,
            clnt->binding_pool, NDR_N_BINDING_POOL);

        if ((clnt->heap = ndr_heap_create()) == NULL)
                goto nomem;

        /* success! */
        bzero(handle, sizeof (*handle));
        handle->clnt = clnt;
        return (0);

nomem:
        free(clnt);
        return (ENOMEM);
}


/*
 * This call must be made to initialize an RPC client structure and bind
 * to the remote service before any RPCs can be exchanged with that service.
 *
 * The mlrpc_handle_t is a wrapper that is used to associate an RPC handle
 * with the client context for an instance of the interface.  The handle
 * is zeroed to ensure that it doesn't look like a valid handle -
 * handle content is provided by the remove service.
 *
 * The client points to this top-level handle so that we know when to
 * unbind and teardown the connection.  As each handle is initialized it
 * will inherit a reference to the client context.
 *
 *
 * Similar to MSRPC RpcBindingBind()
 *
```

```
 * Returns 0 or an NT_STATUS:           (failed in...)
 *
 *      RPC_NT_SERVER_TOO_BUSY          (open pipe)
 *      RPC_NT_SERVER_UNAVAILABLE       (open pipe)
 *      NT_STATUS_ACCESS_DENIED         (open pipe)
 *      NT_STATUS_INVALID_PARAMETER     (rpc bind)
 *      NT_STATUS_INTERNAL_ERROR        (bad args etc)
 *      NT_STATUS_NO_MEMORY
 */
uint32_t
mlrpc_clh_bind(mlrpc_handle_t *handle, ndr_service_t *svc)
{
        ndr_client_t            *clnt = NULL;
        struct smb_ctx          *ctx = NULL;
        uint32_t                status = 0;
        int                     fd = -1;
        int                     rc, retries;

        if ((clnt = handle->clnt) == NULL)
                return (NT_STATUS_INTERNAL_ERROR);
        if ((ctx = clnt->xa_private) == NULL)
                return (NT_STATUS_INTERNAL_ERROR);
        if (clnt->xa_fd != -1)
                return (NT_STATUS_INTERNAL_ERROR);

        /*
         * Open the named pipe.
         *
         * Sometimes a DC may return NT_STATUS_PIPE_NOT_AVAILABLE for
         * the first few seconds during service auto-start.  The client
         * translates that to EBUSY, so when we see that, wait a bit
         * and retry the open for up to rpc_pipe_open_retries.  If we
         * fail even after retries, return RPC_NT_SERVER_TOO_BUSY,
         * which is how callers of this layer expect that reported.
         * We try up to 10 times, with a 0.5 sec. wait after each
         * BUSY failure, giving a total wait here of 5 sec.
         */
        retries = rpc_pipe_open_retries;
retry_open:
        fd = smb_fh_open(ctx, svc->endpoint, O_RDWR);
        if (fd < 0) {
                rc = errno;
                switch (rc) {
                case EBUSY:
                        if (--retries > 0) {
                                (void) poll(NULL, 0, 500);
                                goto retry_open;
                        }
                        status = RPC_NT_SERVER_TOO_BUSY;
                        break;
                case EACCES:
                        status = NT_STATUS_ACCESS_DENIED;
                        break;
                default:
                        status = RPC_NT_SERVER_UNAVAILABLE;
                        break;
                }
                return (status);
        }

        clnt->xa_fd = fd;

        /* Paranoia, in case of re-bind. */
        bzero(&handle->handle, sizeof (ndr_hdid_t));

        /*
```



```
         * Do the OtW RPC bind.
         */
        rc = ndr_clnt_bind(clnt, svc, &clnt->binding);
        switch (rc) {
        case NDR_DRC_FAULT_OUT_OF_MEMORY:
                status = NT_STATUS_NO_MEMORY;
                break;
        case NDR_DRC_FAULT_API_SERVICE_INVALID:
                /* svc->..._uuid parse errors */
                status = NT_STATUS_INTERNAL_ERROR;
                break;
        default:
                if (NDR_DRC_IS_FAULT(rc)) {
                        status = RPC_NT_PROTOCOL_ERROR;
                        break;
                }
                /* FALLTHROUGH */
        case NDR_DRC_OK:
                status = NT_STATUS_SUCCESS;
        }

        if (status != 0) {
                if (fd != -1)
                        (void) smb_fh_close(fd);
                clnt->xa_fd = -1;
        }

        return (status);
}

/*
 * Unbind and close the pipe to an RPC service.
 *
 * Similar to MSRPC RpcBindingUnbind()
 * This should be called after a dropped connection.
 */
void
mlrpc_clh_unbind(mlrpc_handle_t *handle)
{
        ndr_client_t *clnt = handle->clnt;

        if (clnt->xa_fd != -1) {
                (void) smb_fh_close(clnt->xa_fd);
                clnt->xa_fd = -1;
        }
}

/*
 * If the heap has been preserved we need to go through an xa release.
 * The heap is preserved during an RPC call because that's where data
 * returned from the server is stored.
 *
 * Otherwise we destroy the heap directly.
 *
 * Returns the xa_private pointer (if non-NULL) to inform the caller
 * that it can now be destroyed.
 */
void *
mlrpc_clh_free(mlrpc_handle_t *handle)
{
        ndr_client_t *clnt = handle->clnt;
        void *private;

        if (clnt == NULL)
                return (NULL);
```

```
260         /*
261          * Should never get an unbind on inherited handles.
262          * Callers of ndr_inherit_handle() check handles
263          * with ndr_is_bind_handle() before calling this.
264          *
265          * Maybe make this function more tolerant?
266          */
267         assert(handle->clnt->handle == &handle->handle);

269         mlrpc_clh_unbind(handle);

271         if (clnt->heap_preserved)
272                 ndr_clnt_free_heap(clnt); /* xa_release */
273         else
274                 ndr_heap_destroy(clnt->heap);

276         /*
277          * Note: Caller will free the smb_ctx stored in
278          * clnt->xa_private (or possibly reuse it).
279          */
280         private = clnt->xa_private;
281         free(clnt);
282         bzero(handle, sizeof (*handle));
283         return (private);
284 }

286 /*
287  * Call the RPC function identified by opnum.  The remote service is
288  * identified by the handle, which should have been initialized by
289  * ndr_rpc_bind.
290  *
291  * If the RPC call is successful (returns 0), the caller must call
292  * ndr_rpc_release to release the heap.  Otherwise, we release the
293  * heap here.
294  */
295 int
296 ndr_rpc_call(mlrpc_handle_t *handle, int opnum, void *params)
297 {
298         ndr_client_t *clnt = handle->clnt;
299         int rc;

301         if (ndr_rpc_get_heap(handle) == NULL)
302                 return (-1);

304         rc = ndr_clnt_call(clnt->binding, opnum, params);

306         /*
307          * Always clear the nonull flag to ensure
308          * it is not applied to subsequent calls.
309          */
310         clnt->nonull = B_FALSE;

312         if (NDR_DRC_IS_FAULT(rc)) {
313                 ndr_rpc_release(handle);
314                 return (-1);
315         }

317         return (0);
318 }

320 /*
321  * Outgoing strings should not be null terminated.
322  */
323 void
324 ndr_rpc_set_nonull(mlrpc_handle_t *handle)
325 {
```



```
326         handle->clnt->nonull = B_TRUE;
327 }

329 /*
330  * Get the session key from a bound RPC client handle.
331  *
332  * The key returned is the 16-byte "user session key"
333  * established by the underlying authentication protocol
334  * (either Kerberos or NTLM).  This key is needed for
335  * SAM RPC calls such as SamrSetInformationUser, etc.
336  * See [MS-SAMR] sections: 2.2.3.3, 2.2.7.21, 2.2.7.25.
337  *
338  * Returns zero (success) or an errno.
339  */
340 int
341 ndr_rpc_get_ssnkey(mlrpc_handle_t *handle, uchar_t *key, size_t len)
342 {
343         ndr_client_t *clnt = handle->clnt;

345         if (clnt == NULL || clnt->xa_fd == -1)
346                 return (EINVAL);

348         return (smb_fh_getssnkey(clnt->xa_fd, key, len));
349 }

351 void *
352 ndr_rpc_malloc(mlrpc_handle_t *handle, size_t size)
353 {
354         ndr_heap_t *heap;

356         if ((heap = ndr_rpc_get_heap(handle)) == NULL)
357                 return (NULL);

359         return (ndr_heap_malloc(heap, size));
360 }

362 ndr_heap_t *
363 ndr_rpc_get_heap(mlrpc_handle_t *handle)
364 {
365         ndr_client_t *clnt = handle->clnt;

367         if (clnt->heap == NULL)
368                 clnt->heap = ndr_heap_create();

370         return (clnt->heap);
371 }

373 /*
374  * Must be called by RPC clients to free the heap after a successful RPC
375  * call, i.e. ndr_rpc_call returned 0.  The caller should take a copy
376  * of any data returned by the RPC prior to calling this function because
377  * returned data is in the heap.
378  */
379 void
380 ndr_rpc_release(mlrpc_handle_t *handle)
381 {
382         ndr_client_t *clnt = handle->clnt;

384         if (clnt->heap_preserved)
385                 ndr_clnt_free_heap(clnt);
386         else
387                 ndr_heap_destroy(clnt->heap);

389         clnt->heap = NULL;
390 }
```

```
392 /*
393  * Returns true if the handle is null.
394  * Otherwise returns false.
395  */
396 boolean_t
397 ndr_is_null_handle(mlrpc_handle_t *handle)
398 {
399         static const ndr_hdid_t hdid0 = {0};

401         if (handle == NULL || handle->clnt == NULL)
402                 return (B_TRUE);

404         if (!memcmp(&handle->handle, &hdid0, sizeof (hdid0)))
405                 return (B_TRUE);

407         return (B_FALSE);
408 }

410 /*
411  * Returns true if the handle is the top level bind handle.
412  * Otherwise returns false.
413  */
414 boolean_t
415 ndr_is_bind_handle(mlrpc_handle_t *handle)
416 {
417         return (handle->clnt->handle == &handle->handle);
418 }

420 /*
421  * Pass the client reference from parent to child.
422  */
423 void
424 ndr_inherit_handle(mlrpc_handle_t *child, mlrpc_handle_t *parent)
425 {
426         child->clnt = parent->clnt;
427 }

429 /*
430  * ndr_rpc_status remains in libmlsvc mlsvc_client.c
431  */

433 /*
434  * The following functions provide the client callback interface.
435  * If the caller hasn't provided a heap, create one here.
436  */
437 static int
438 ndr_xa_init(ndr_client_t *clnt, ndr_xa_t *mxa)
439 {
440         ndr_stream_t *recv_nds = &mxa->recv_nds;
441         ndr_stream_t *send_nds = &mxa->send_nds;
442         ndr_heap_t *heap = clnt->heap;
443         int             rc;

445         if (heap == NULL) {
446                 if ((heap = ndr_heap_create()) == NULL)
447                         return (-1);

449                 clnt->heap = heap;
450         }

452         mxa->heap = heap;

454         rc = nds_initialize(send_nds, 0, NDR_MODE_CALL_SEND, heap);
455         if (rc == 0)
456                 rc = nds_initialize(recv_nds, NDR_PDU_SIZE_HINT_DEFAULT,
457                     NDR_MODE_RETURN_RECV, heap);
```



```
459         if (rc != 0) {
460                 nds_destruct(&mxa->recv_nds);
461                 nds_destruct(&mxa->send_nds);
462                 ndr_heap_destroy(mxa->heap);
463                 mxa->heap = NULL;
464                 clnt->heap = NULL;
465                 return (-1);
466         }

468         if (clnt->nonull)
469                 NDS_SETF(send_nds, NDS_F_NONULL);

471         return (0);
472 }

474 /*
475  * This is the entry pointy for an RPC client call exchange with
476  * a server, which will result in an smbrdr SmbTransact request.
477  *
478  * SmbTransact should return the number of bytes received, which
479  * we record as the PDU size, or a negative error code.
480  */
481 static int
482 ndr_xa_exchange(ndr_client_t *clnt, ndr_xa_t *mxa)
483 {
484         ndr_stream_t *recv_nds = &mxa->recv_nds;
485         ndr_stream_t *send_nds = &mxa->send_nds;
486         int err, more, nbytes;

488         nbytes = recv_nds->pdu_max_size;
489         err = smb_fh_xactnp(clnt->xa_fd,
490             send_nds->pdu_size, (char *)send_nds->pdu_base_offset,
491             &nbytes, (char *)recv_nds->pdu_base_offset, &more);
492         if (err) {
493                 recv_nds->pdu_size = 0;
494                 return (-1);
495         }

497         recv_nds->pdu_size = nbytes;
498         return (0);
499 }

501 /*
502  * This entry point will be invoked if the xa-exchange response contained
503  * only the first fragment of a multi-fragment response.  The RPC client
504  * code will then make repeated xa-read requests to obtain the remaining
505  * fragments, which will result in smbrdr SmbReadX requests.
506  *
507  * SmbReadX should return the number of bytes received, in which case we
508  * expand the PDU size to include the received data, or a negative error
509  * code.
510  */
511 static int
512 ndr_xa_read(ndr_client_t *clnt, ndr_xa_t *mxa)
513 {
514         ndr_stream_t *nds = &mxa->recv_nds;
515         int len;
516         int nbytes;

518         if ((len = (nds->pdu_max_size - nds->pdu_size)) < 0)
519                 return (-1);

521         nbytes = smb_fh_read(clnt->xa_fd, 0, len,
522             (char *)nds->pdu_base_offset + nds->pdu_size);
```

```
524         if (nbytes < 0)
525                 return (-1);

527         nds->pdu_size += nbytes;

529         if (nds->pdu_size > nds->pdu_max_size) {
530                 nds->pdu_size = nds->pdu_max_size;
531                 return (-1);
532         }

534         return (nbytes);
535 }

537 /*
538  * Preserve the heap so that the client application has access to data
539  * returned from the server after an RPC call.
540  */
541 static void
542 ndr_xa_preserve(ndr_client_t *clnt, ndr_xa_t *mxa)
543 {
544         assert(clnt->heap == mxa->heap);

546         clnt->heap_preserved = B_TRUE;
547         mxa->heap = NULL;
548 }

550 /*
551  * Dispose of the transaction streams.  If the heap has not been
552  * preserved, we can destroy it here.
553  */
554 static void
555 ndr_xa_destruct(ndr_client_t *clnt, ndr_xa_t *mxa)
556 {
557         nds_destruct(&mxa->recv_nds);
558         nds_destruct(&mxa->send_nds);

560         if (!clnt->heap_preserved) {
561                 ndr_heap_destroy(mxa->heap);
562                 mxa->heap = NULL;
563                 clnt->heap = NULL;
564         }
565 }

567 /*
568  * Dispose of a preserved heap.
569  */
570 static void
571 ndr_xa_release(ndr_client_t *clnt)
572 {
573         if (clnt->heap_preserved) {
574                 ndr_heap_destroy(clnt->heap);
575                 clnt->heap = NULL;
576                 clnt->heap_preserved = B_FALSE;
577         }
578 }
```

```
**********************************************************
   16870 Sun Mar 18 01:12:58 2018
new/usr/src/lib/libmlrpc/common/ndr.h
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  23  * Use is subject to license terms.
  24  *
  25  * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
  26  */

  28 #ifndef _SMBSRV_NDR_H
  29 #define _SMBSRV_NDR_H

  31 /*
  32  * Network Data Representation (NDR) is a compatible subset of DCE RPC
  33  * and MSRPC NDR.  NDR is used to move parameters consisting of
  34  * complicated trees of data constructs between an RPC client and server.
  35  *
  36  * CAE Specification (1997)
  37  * DCE 1.1: Remote Procedure Call
  38  * Document Number: C706
  39  * The Open Group
  40  * ogspecs@opengroup.org
  41  */

  43 #if defined(_KERNEL) || defined(_FAKE_KERNEL)
  44 #error "not used in kernel code"
  45 #else /* _KERNEL */
  43 #include <sys/types.h>
  44 #include <sys/uio.h>
  48 #include <syslog.h>
  45 #include <stdlib.h>
  46 #include <string.h>
  51 #include <smbsrv/wintypes.h>
  52 #include <smbsrv/ndl/rpcpdu.ndl>
  53 #include <smbsrv/string.h>
  54 #endif  /* _KERNEL */

  48 #include <smb/wintypes.h>
  49 #include <libmlrpc/ndrtypes.ndl>
  50 #include <libmlrpc/rpcpdu.ndl>

  52 #ifdef __cplusplus
  53 extern "C" {
```



```
  54 #endif

  56 /*
  57  * Normal sequence:
  58  *      - Application calls client-side stub w/ TOP-MOST arg structure
  59  *      - client stub performs NDR_M_OP_MARSHALL+NDR_DIR_IN
  60  *      - PDU conveyed (request, aka call, aka query)
  61  *      - server stub performs NDR_M_OP_UNMARSHALL+NDR_DIR_IN
  62  *      - server function called w/ TOP-MOST arg structure
  63  *      - server function returns w/ TOP-MOST arg structure modified
  64  *      - server stub performs NDR_M_OP_MARSHALL+NDR_DIR_OUT
  65  *      - PDU conveyed (reply, aka result, aka response)
  66  *      - client stub performs NDR_M_OP_UNMARSHALL+NDR_DIR_OUT
  67  *      - return to Application w/ TOP-MOST arg structure modified
  68  *
  69  * An interface is a sequence of top-most constructs.  Each top-most
  70  * construct corresponds to one parameter, either argument or return
  71  * value.
  72  *
  73  * A top-most construct is a sequence of outer constructs.  The first
  74  * outer construct is the referent of the argument, and the subsequent
  75  * outer constructs are descendents referenced by pointers from prior
  76  * constructs.
  77  *
  78  * An outer construct is a sequence of variable-sized info, fixed-sized
  79  * data, and variable-sized data.
  80  */

  82 /*
  83  * Terminology
  84  *
  85  * The ALL UPPER CASE terms recur in the DCE/RPC documentation.
  86  * The mixed-case names have been introduced as a reading aid.
  87  *
  88  * Size         The size of an array in elements. Think of this
  89  *              as the amount to malloc().
  90  *
  91  * Length       The number of elements of an array which are significant
  92  *              Think of this as the amount to bcopy().
  93  *
  94  * Known        Size/length is known at build time.
  95  *
  96  * Determined   Size/length is determined at run time.
  97  *
  98  * FIXED        The Size and Length are Known.
  99  *              Think of this as a string constant or a DOS 8.3 file name.
 100  *              char array[] = "A Constant Size/Length";
 101  *
 102  * CONFORMANT   The Size is Determined. Length is the same as Size.
 103  *              Think of this as strdup().
 104  *              char *array = strdup("Something");
 105  *
 106  * VARYING      The Size is Known. The Length is determined.
 107  *              Think of this as a strcpy() of a variable length string
 108  *              into a fixed length buffer:
 109  *              char array[100];
 110  *              strcpy(array, "very short string");
 111  *
 112  * VARYING/CONFORMANT
 113  *              The Size is Determined. The Length is separately Determined.
 114  *              Think of this like:
 115  *              char *array = malloc(size);
 116  *              strcpy(array, "short string");
 117  *
 118  * STRING       Strings can be CONFORMANT, VARYING, or CONFORMANT/VARYING.
 119  *              A string is fundamentally an array with the last
```

```
 120  *              significant element some sort of NULL.
 121  */

 123 #define NDR_F_NONE              0x0000  /* no flags */
 124 #define NDR_F_PARAMS_MASK       0x00FF
 125 #define NDR_F_SIZE_IS           0x0001  /* [size_is(X)] required/given */
 126 #define NDR_F_LENGTH_IS         0x0002  /* not implemented */
 127 #define NDR_F_SWITCH_IS         0x0004  /* [switch_is(X)] req./given */
 128 #define NDR_F_IS_STRING         0x0008  /* [string] req./given */
 129 #define NDR_F_IS_POINTER        0x0010  /* TYPE * ... req./given */
 130 #define NDR_F_IS_REFERENCE      0x0020  /* TYPE & ... req./given */
 131 #define NDR_F_DIMENSION_IS      0x0040  /* TYPE [N] req./given */

 133 #define NDR_F_WHENCE_MASK       0x00F0
 134 #define NDR_F_BACKPTR           0x0010  /* ref cause by pointer */
 135 #define NDR_F_OUTER             0x0020  /* ref caused by outer */
 136 #define NDR_F_TOPMOST           0x0040  /* ref caused by topmost */

 138 #define NDR_F_TYPEOP_MASK       0x0F00
 139 #define NDR_F_ARRAY             0x0100  /* type is array of somethings */
 140 #define NDR_F_POINTER           0x0200  /* type is pointer to something(s) */
 141 #define NDR_F_STRING            0x0300  /* type is string of somethings */
 142 #define NDR_F_UNION             0x0400  /* type is a union */
 143 #define NDR_F_STRUCT            0x0500  /* type is a structure */
 144 #define NDR_F_OPERATION         0x0600  /* type is a structure, special */
 145 #define NDR_F_INTERFACE         0x0700  /* type is a union, special */
 146 #define NDR_F_CONFORMANT        0x1000  /* struct conforming (var-size tail) */
 147 #define NDR_F_VARYING           0x2000  /* not implemented */

 149 struct ndr_heap;
 150 struct ndr_stream;
 151 struct ndr_reference;

 153 typedef uint16_t ndr_wchar_t;

 155 typedef struct ndr_typeinfo {
 156         unsigned char           version;        /* sanity check */
 157         unsigned char           alignment;      /* mask */
 158         unsigned short          type_flags;     /* NDR_F_... */
 159         int                     (*ndr_func)(struct ndr_reference *);
 160         unsigned short          pdu_size_fixed_part;
 161         unsigned short          pdu_size_variable_part;
 162         unsigned short          c_size_fixed_part;
 163         unsigned short          c_size_variable_part;
 164 } ndr_typeinfo_t;
_____unchanged_portion_omitted_

 255 #define NDR_M_OP_NONE           0x00
 256 #define NDR_M_OP_MARSHALL       0x01    /* data moving from datum to PDU */
 257 #define NDR_M_OP_UNMARSHALL     0x02    /* data moving from PDU to datum */

 259 #define NDR_DIR_NONE            0x00
 260 #define NDR_DIR_IN              0x10    /* data moving from caller to callee */
 261 #define NDR_DIR_OUT             0x20    /* data moving from callee to caller */

 263 #define NDR_MODE_CALL_SEND      (NDR_M_OP_MARSHALL + NDR_DIR_IN)
 264 #define NDR_MODE_CALL_RECV      (NDR_M_OP_UNMARSHALL + NDR_DIR_IN)
 265 #define NDR_MODE_RETURN_SEND    (NDR_M_OP_MARSHALL + NDR_DIR_OUT)
 266 #define NDR_MODE_RETURN_RECV    (NDR_M_OP_UNMARSHALL + NDR_DIR_OUT)
 267 #define NDR_MODE_BUF_ENCODE     NDR_MODE_CALL_SEND
 268 #define NDR_MODE_BUF_DECODE     NDR_MODE_RETURN_RECV

 270 #define NDR_MODE_TO_M_OP(MODE)  ((MODE) & 0x0F)
 271 #define NDR_MODE_TO_DIR(MODE)   ((MODE) & 0xF0)
 272 #define NDR_M_OP_AND_DIR_TO_MODE(M_OP, DIR)     ((M_OP)|(DIR))
```



```
 274 #define NDR_MODE_MATCH(NDS, MODE) \
 275         (NDR_M_OP_AND_DIR_TO_MODE((NDS)->m_op, (NDS)->dir) == (MODE))

 277 #define NDR_IS_FIRST_FRAG(F)    ((F) & NDR_PFC_FIRST_FRAG)
 278 #define NDR_IS_LAST_FRAG(F)     ((F) & NDR_PFC_LAST_FRAG)
 279 #define NDR_IS_SINGLE_FRAG(F)   \
 280         (NDR_IS_FIRST_FRAG((F)) && NDR_IS_LAST_FRAG((F)))

 282 #define NDS_F_NONE              0x00
 283 #define NDS_F_NOTERM            0x01    /* strings are not null terminated */
 284 #define NDS_F_NONULL            0x02    /* strings: no null on size_is */
 285 #define NDS_SETF(S, F)          ((S)->flags |= (F))
 286 #define NDS_CLEARF(S, F)        ((S)->flags &= ~(F))

 288 #define NDR_ERR_MALLOC_FAILED           -1
 289 #define NDR_ERR_M_OP_INVALID            -2
 290 #define NDR_ERR_UNDERFLOW               -3
 291 #define NDR_ERR_GROW_FAILED             -4      /* overflow */
 292 #define NDR_ERR_PAD_FAILED              -5      /* couldn't possibly happen */
 293 #define NDR_ERR_OUTER_HEADER_BAD        -6
 294 #define NDR_ERR_SWITCH_VALUE_ILLEGAL    -7
 295 #define NDR_ERR_SWITCH_VALUE_INVALID    -8
 296 #define NDR_ERR_SWITCH_VALUE_MISSING    -9
 297 #define NDR_ERR_SIZE_IS_MISMATCH_PDU    -10
 298 #define NDR_ERR_SIZE_IS_MISMATCH_AFTER  -11
 299 #define NDR_ERR_SIZE_IS_UNEXPECTED      -12
 300 #define NDR_ERR_SIZE_IS_DUPLICATED      -13
 301 #define NDR_ERR_OUTER_PARAMS_MISMATCH   -14
 302 #define NDR_ERR_ARRAY_VARLEN_ILLEGAL    -15
 303 #define NDR_ERR_ARRAY_UNION_ILLEGAL     -16
 304 #define NDR_ERR_OUTER_PARAMS_BAD        -17
 305 #define NDR_ERR_OUTER_UNION_ILLEGAL     -18
 306 #define NDR_ERR_TOPMOST_UNION_ILLEGAL   -19
 307 #define NDR_ERR_TOPMOST_VARLEN_ILLEGAL  -20
 308 #define NDR_ERR_INNER_PARAMS_BAD        -21
 309 #define NDR_ERR_UNIMPLEMENTED           -22
 310 #define NDR_ERR_NOT_AN_INTERFACE        -23
 311 #define NDR_ERR_STRLEN                  -24
 312 #define NDR_ERR_STRING_SIZING           -25
 313 #define NDR_ERR_BOUNDS_CHECK            -26

 315 #define NDR_SET_ERROR(REF, ERROR)                       \
 316         ((REF)->stream->error = (ERROR),                \
 317         (REF)->stream->error_ref = __LINE__,            \
 318         NDS_TATTLE_ERROR((REF)->stream, 0, REF))

 320 #define NDR_TATTLE(REF, WHAT) \
 321         (*(REF)->stream->ndo->ndo_tattle)((REF)->stream, WHAT, REF)

 323 #define MEMBER_STR(MEMBER) #MEMBER

 325 #define NDR_DIR_IS_IN  (encl_ref->stream->dir == NDR_DIR_IN)
 326 #define NDR_DIR_IS_OUT (encl_ref->stream->dir == NDR_DIR_OUT)

 328 #define NDR_MEMBER_WITH_ARG(TYPE, MEMBER, OFFSET, \
 329                 ARGFLAGS, ARGMEM, ARGVAL) { \
 330                 myref.pdu_offset = encl_ref->pdu_offset + (OFFSET);     \
 331                 myref.name = MEMBER_STR(MEMBER);                        \
 332                 myref.datum = (char *)&val->MEMBER;                     \
 333                 myref.inner_flags = ARGFLAGS;                           \
 334                 myref.ti = &ndt_##TYPE;                                 \
 335                 myref.ARGMEM = ARGVAL;                                  \
 336                 if (!ndr_inner(&myref))                                 \
 337                         return (0);                                     \
 338         }
```

```
340 #define NDR_MEMBER(TYPE, MEMBER, OFFSET) \
341         NDR_MEMBER_WITH_ARG(TYPE, MEMBER, OFFSET, \
342                 NDR_F_NONE, size_is, 0)

344 #define NDR_MEMBER_ARR_WITH_SIZE_IS(TYPE, MEMBER, OFFSET, SIZE_IS) \
345         NDR_MEMBER_WITH_ARG(TYPE, MEMBER, OFFSET, \
346                 NDR_F_SIZE_IS, size_is, SIZE_IS)

348 #define NDR_MEMBER_ARR_WITH_DIMENSION(TYPE, MEMBER, OFFSET, SIZE_IS) \
349         NDR_MEMBER_WITH_ARG(TYPE, MEMBER, OFFSET, \
350                 NDR_F_DIMENSION_IS, dimension_is, SIZE_IS)

352 #define NDR_MEMBER_PTR_WITH_SIZE_IS(TYPE, MEMBER, OFFSET, SIZE_IS) \
353         NDR_MEMBER_WITH_ARG(TYPE, MEMBER, OFFSET, \
354                 NDR_F_SIZE_IS+NDR_F_IS_POINTER, size_is, SIZE_IS)

356 #define NDR_MEMBER_PTR(TYPE, MEMBER, OFFSET)            \
357         NDR_MEMBER_WITH_ARG(TYPE, MEMBER, OFFSET,       \
358                 NDR_F_IS_POINTER, size_is, 0)

360 #define NDR_MEMBER_WITH_SWITCH_IS(TYPE, MEMBER, OFFSET, SWITCH_IS)      \
361         NDR_MEMBER_WITH_ARG(TYPE, MEMBER, OFFSET,                       \
362                 NDR_F_SWITCH_IS, switch_is, SWITCH_IS)


365 #define NDR_TOPMOST_MEMBER_WITH_ARG(TYPE, MEMBER, \
366                 ARGFLAGS, ARGMEM, ARGVAL) { \
367                 myref.pdu_offset = -1;                                  \
368                 myref.name = MEMBER_STR(MEMBER);                        \
369                 myref.datum = (char *)&val->MEMBER;                     \
370                 myref.inner_flags = ARGFLAGS;                           \
371                 myref.ti = &ndt_##TYPE;                                 \
372                 myref.ARGMEM = ARGVAL;                                  \
373                 if (!ndr_topmost(&myref))                               \
374                         return (0);                                     \
375         }

377 #define NDR_TOPMOST_MEMBER(TYPE, MEMBER)                                \
378         NDR_TOPMOST_MEMBER_WITH_ARG(TYPE, MEMBER,                       \
379                 NDR_F_NONE, size_is, 0)

381 #define NDR_TOPMOST_MEMBER_ARR_WITH_SIZE_IS(TYPE, MEMBER, SIZE_IS)      \
382         NDR_TOPMOST_MEMBER_WITH_ARG(TYPE, MEMBER,                       \
383                 NDR_F_SIZE_IS, size_is, SIZE_IS)

385 #define NDR_TOPMOST_MEMBER_ARR_WITH_DIMENSION(TYPE, MEMBER, SIZE_IS)    \
386         NDR_TOPMOST_MEMBER_WITH_ARG(TYPE, MEMBER,                       \
387                 NDR_F_DIMENSION_IS, dimension_is, SIZE_IS)

389 #define NDR_TOPMOST_MEMBER_PTR_WITH_SIZE_IS(TYPE, MEMBER, SIZE_IS)      \
390         NDR_TOPMOST_MEMBER_WITH_ARG(TYPE, MEMBER,                       \
391                 NDR_F_SIZE_IS+NDR_F_IS_POINTER, size_is, SIZE_IS)

393 #define NDR_TOPMOST_MEMBER_PTR(TYPE, MEMBER)            \
394         NDR_TOPMOST_MEMBER_WITH_ARG(TYPE, MEMBER,       \
395                 NDR_F_IS_POINTER, size_is, 0)

397 #define NDR_TOPMOST_MEMBER_REF(TYPE, MEMBER)            \
398         NDR_TOPMOST_MEMBER_WITH_ARG(TYPE, MEMBER,       \
399                 NDR_F_IS_REFERENCE, size_is, 0)

401 #define NDR_TOPMOST_MEMBER_REF_WITH_SIZE_IS(TYPE, MEMBER, SIZE_IS)      \
402         NDR_TOPMOST_MEMBER_WITH_ARG(TYPE, MEMBER,                       \
403                 NDR_F_SIZE_IS+NDR_F_IS_REFERENCE, size_is, SIZE_IS)

405 #define NDR_TOPMOST_MEMBER_WITH_SWITCH_IS(TYPE, MEMBER, SWITCH_IS)      \
```



```
406         NDR_TOPMOST_MEMBER_WITH_ARG(TYPE, MEMBER,                       \
407                 NDR_F_SWITCH_IS, switch_is, SWITCH_IS)

409 /* this is assuming offset+0 */
410 #define NDR_PARAMS_MEMBER_WITH_ARG(TYPE, MEMBER, ARGFLAGS, \
411         ARGMEM, ARGVAL) { \
412                 myref.pdu_offset = encl_ref->pdu_offset;                \
413                 myref.name = MEMBER_STR(MEMBER);                        \
414                 myref.datum = (char *)&val->MEMBER;                     \
415                 myref.inner_flags = ARGFLAGS;                           \
416                 myref.ti = &ndt_##TYPE;                                 \
417                 myref.ARGMEM = ARGVAL;                                  \
418                 if (!ndr_params(&myref))                                \
419                         return (0);                                     \
420         }

422 #define NDR_PARAMS_MEMBER(TYPE, MEMBER)                 \
423         NDR_PARAMS_MEMBER_WITH_ARG(TYPE, MEMBER,        \
424         NDR_F_NONE, size_is, 0)

426 #define NDR_STRING_DIM          1
427 #define NDR_ANYSIZE_DIM         1

429 int ndo_process(struct ndr_stream *, ndr_typeinfo_t *, char *);
430 int ndo_operation(struct ndr_stream *, ndr_typeinfo_t *, int opnum, char *);
431 void ndo_printf(struct ndr_stream *, ndr_ref_t *, const char *, ...);
432 void ndo_trace(const char *);
433 void ndo_fmt(struct ndr_stream *, ndr_ref_t *, char *);

435 int ndr_params(ndr_ref_t *);
436 int ndr_topmost(ndr_ref_t *);
437 int ndr_run_outer_queue(struct ndr_stream *);
438 int ndr_outer(ndr_ref_t *);
439 int ndr_outer_fixed(ndr_ref_t *);
440 int ndr_outer_fixed_array(ndr_ref_t *);
441 int ndr_outer_conformant_array(ndr_ref_t *);
442 int ndr_outer_conformant_construct(ndr_ref_t *);
443 int ndr_size_is(ndr_ref_t *);
444 int ndr_outer_string(ndr_ref_t *);
445 int ndr_outer_peek_sizing(ndr_ref_t *, unsigned, unsigned long *);
446 int ndr_outer_poke_sizing(ndr_ref_t *, unsigned, unsigned long *);
447 int ndr_outer_align(ndr_ref_t *);
448 int ndr_outer_grow(ndr_ref_t *, unsigned);
449 int ndr_inner(ndr_ref_t *);
450 int ndr_inner_pointer(ndr_ref_t *);
451 int ndr_inner_reference(ndr_ref_t *);
452 int ndr_inner_array(ndr_ref_t *);

454 size_t ndr_mbstowcs(struct ndr_stream *, ndr_wchar_t *, const char *, size_t);
458 size_t ndr_mbstowcs(struct ndr_stream *, smb_wchar_t *, const char *, size_t);
459 int ndr_mbtowc(struct ndr_stream *, smb_wchar_t *, const char *, size_t);

456 void nds_bswap(void *src, void *dst, size_t len);

458 #ifdef __cplusplus
459 }
```

_____unchanged_portion_omitted_

```
**********************************************************
    8387 Sun Mar 18 01:12:58 2018
new/usr/src/lib/libmlrpc/common/ndr_client.c
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 */

#include <sys/errno.h>
#include <string.h>
#include <strings.h>

#include <libmlrpc.h>
#include <smbsrv/libsmb.h>
#include <smbsrv/libmlrpc.h>

#define NDR_DEFAULT_FRAGSZ      8192
#define NDR_MULTI_FRAGSZ        (60 * 1024)

static void ndr_clnt_init_hdr(ndr_client_t *, ndr_xa_t *);
static int ndr_clnt_get_frags(ndr_client_t *, ndr_xa_t *);
static int ndr_clnt_get_frag(ndr_client_t *, ndr_xa_t *, ndr_common_header_t *);

int
ndr_clnt_bind(ndr_client_t *clnt, ndr_service_t *msvc,
ndr_clnt_bind(ndr_client_t *clnt, const char *service_name,
    ndr_binding_t **ret_binding_p)
{
        ndr_service_t           *msvc;
        ndr_binding_t           *mbind;
        ndr_xa_t                mxa;
        ndr_bind_hdr_t          *bhdr;
        ndr_p_cont_elem_t       *pce;
        ndr_bind_ack_hdr_t      *bahdr;
        ndr_p_result_t          *pre;
        int                     rc;

        bzero(&mxa, sizeof (mxa));

        msvc = ndr_svc_lookup_name(service_name);
        if (msvc == NULL)
                return (NDR_DRC_FAULT_API_SERVICE_INVALID);
```



```
        mxa.binding_list = clnt->binding_list;
        if ((mbind = ndr_svc_new_binding(&mxa)) == NULL)
                return (NDR_DRC_FAULT_API_BIND_NO_SLOTS);

        ndr_clnt_init_hdr(clnt, &mxa);

        bhdr = &mxa.send_hdr.bind_hdr;
        bhdr->common_hdr.ptype = NDR_PTYPE_BIND;
        bhdr->common_hdr.frag_length = sizeof (*bhdr);
        bhdr->max_xmit_frag = NDR_DEFAULT_FRAGSZ;
        bhdr->max_recv_frag = NDR_DEFAULT_FRAGSZ;
        bhdr->assoc_group_id = 0;
        bhdr->p_context_elem.n_context_elem = 1;

        /* Assign presentation context id */
        pce = &bhdr->p_context_elem.p_cont_elem[0];
        pce->p_cont_id = clnt->next_p_cont_id++;
        pce->n_transfer_syn = 1;

        /* Set up UUIDs and versions from the service */
        pce->abstract_syntax.if_version = msvc->abstract_syntax_version;
        rc = ndr_uuid_parse(msvc->abstract_syntax_uuid,
            &pce->abstract_syntax.if_uuid);
        if (rc != 0)
                return (NDR_DRC_FAULT_API_SERVICE_INVALID);

        pce->transfer_syntaxes[0].if_version = msvc->transfer_syntax_version;
        rc = ndr_uuid_parse(msvc->transfer_syntax_uuid,
            &pce->transfer_syntaxes[0].if_uuid);
        if (rc != 0)
                return (NDR_DRC_FAULT_API_SERVICE_INVALID);

        /* Format and exchange the PDU */

        if ((*clnt->xa_init)(clnt, &mxa) < 0)
                return (NDR_DRC_FAULT_OUT_OF_MEMORY);

        rc = ndr_encode_pdu_hdr(&mxa);
        if (NDR_DRC_IS_FAULT(rc))
                goto fault_exit;

        if ((*clnt->xa_exchange)(clnt, &mxa) < 0) {
                rc = NDR_DRC_FAULT_SEND_FAILED;
                goto fault_exit;
        }

        rc = ndr_decode_pdu_hdr(&mxa);
        if (NDR_DRC_IS_FAULT(rc))
                goto fault_exit;

        /* done with buffers */
        (*clnt->xa_destruct)(clnt, &mxa);

        bahdr = &mxa.recv_hdr.bind_ack_hdr;

        if (mxa.ptype != NDR_PTYPE_BIND_ACK)
                return (NDR_DRC_FAULT_RECEIVED_MALFORMED);

        if (bahdr->p_result_list.n_results != 1)
                return (NDR_DRC_FAULT_RECEIVED_MALFORMED);

        pre = &bahdr->p_result_list.p_results[0];

        if (pre->result != NDR_PCDR_ACCEPTANCE)
                return (NDR_DRC_FAULT_RECEIVED_MALFORMED);
```

```
121         mbind->p_cont_id = pce->p_cont_id;
122         mbind->which_side = NDR_BIND_SIDE_CLIENT;
123         mbind->clnt = clnt;
124         mbind->service = msvc;
125         mbind->instance_specific = 0;

127         *ret_binding_p = mbind;
128         return (NDR_DRC_OK);

130 fault_exit:
131         (*clnt->xa_destruct)(clnt, &mxa);
132         return (rc);
133 }

135 int
136 ndr_clnt_call(ndr_binding_t *mbind, int opnum, void *params)
137 {
138         ndr_client_t            *clnt = mbind->clnt;
143         ndr_service_t           *msvc = mbind->service;
139         ndr_xa_t                mxa;
140         ndr_request_hdr_t       *reqhdr;
141         ndr_common_header_t     *rsphdr;
142         unsigned long           recv_pdu_scan_offset;
143         int                     rc;

150         if (ndr_svc_lookup_name(msvc->name) == NULL)
151                 return (NDR_DRC_FAULT_API_SERVICE_INVALID);

145         bzero(&mxa, sizeof (mxa));
146         mxa.ptype = NDR_PTYPE_REQUEST;
147         mxa.opnum = opnum;
148         mxa.binding = mbind;

150         ndr_clnt_init_hdr(clnt, &mxa);

152         reqhdr = &mxa.send_hdr.request_hdr;
153         reqhdr->common_hdr.ptype = NDR_PTYPE_REQUEST;
154         reqhdr->p_cont_id = mbind->p_cont_id;
155         reqhdr->opnum = opnum;

157         rc = (*clnt->xa_init)(clnt, &mxa);
158         if (NDR_DRC_IS_FAULT(rc))
159                 return (rc);

161         /* Reserve room for hdr */
162         mxa.send_nds.pdu_scan_offset = sizeof (*reqhdr);

164         rc = ndr_encode_call(&mxa, params);
165         if (!NDR_DRC_IS_OK(rc))
166                 goto fault_exit;

168         mxa.send_nds.pdu_scan_offset = 0;

170         /*
171          * Now we have the PDU size, we need to set up the
172          * frag_length and calculate the alloc_hint.
173          */
174         mxa.send_hdr.common_hdr.frag_length = mxa.send_nds.pdu_size;
175         reqhdr->alloc_hint = mxa.send_nds.pdu_size -
176             sizeof (ndr_request_hdr_t);

178         rc = ndr_encode_pdu_hdr(&mxa);
179         if (NDR_DRC_IS_FAULT(rc))
180                 goto fault_exit;
```



```
182         rc = (*clnt->xa_exchange)(clnt, &mxa);
183         if (NDR_DRC_IS_FAULT(rc))
184                 goto fault_exit;

186         rc = ndr_decode_pdu_hdr(&mxa);
187         if (NDR_DRC_IS_FAULT(rc))
188                 goto fault_exit;

190         if (mxa.ptype != NDR_PTYPE_RESPONSE) {
191                 rc = NDR_DRC_FAULT_RECEIVED_MALFORMED;
192                 goto fault_exit;
193         }

195         rsphdr = &mxa.recv_hdr.common_hdr;

197         if (!NDR_IS_LAST_FRAG(rsphdr->pfc_flags)) {
198                 /*
199                  * This is a multi-fragment response.
200                  * Preserve the current scan offset while getting
201                  * fragments so that we can continue afterward
202                  * as if we had received the entire response as
203                  * a single PDU.
204                  */
205                 (void) NDS_GROW_PDU(&mxa.recv_nds, NDR_MULTI_FRAGSZ, NULL);

207                 recv_pdu_scan_offset = mxa.recv_nds.pdu_scan_offset;
208                 mxa.recv_nds.pdu_scan_offset = rsphdr->frag_length;
209                 mxa.recv_nds.pdu_size = rsphdr->frag_length;

211                 if (ndr_clnt_get_frags(clnt, &mxa) < 0) {
212                         rc = NDR_DRC_FAULT_RECEIVED_MALFORMED;
213                         goto fault_exit;
214                 }

216                 mxa.recv_nds.pdu_scan_offset = recv_pdu_scan_offset;
217         }

219         rc = ndr_decode_return(&mxa, params);
220         if (NDR_DRC_IS_FAULT(rc))
221                 goto fault_exit;

223         (*clnt->xa_preserve)(clnt, &mxa);
224         (*clnt->xa_destruct)(clnt, &mxa);
225         return (NDR_DRC_OK);

227 fault_exit:
228         (*clnt->xa_destruct)(clnt, &mxa);
229         return (rc);
230 }
```

*unchanged_portion_omitted_*

****
7712 Sun Mar 18 01:12:58 2018
new/usr/src/lib/libmlrpc/common/ndr_heap.c
1575 untangle libmlrpc ... (libmlrpc)
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 */

/*
 * NDR heap management. The heap is used for temporary storage by
 * both the client and server side library routines.  In order to
 * support the different requirements of the various RPCs, the heap
 * can grow dynamically if required.  We start with a single block
 * and perform sub-allocations from it.  If an RPC requires more space
 * we will continue to add it a block at a time.  This means that we
 * don't hog lots of memory on every call to support the few times
 * that we actually need a lot heap space.
 *
 * Note that there is no individual free function.  Once space has been
 * allocated, it remains allocated until the heap is destroyed.  This
 * shouldn't be an issue because the heap is being filled with data to
 * be marshalled or unmarshalled and we need it all to be there until
 * the point that the entire heap is no longer required.
 */

#include <sys/errno.h>
#include <stdlib.h>
#include <string.h>
#include <strings.h>
#include <sys/uio.h>

#include <libmlrpc.h>
#include <ndr_wchar.h>
#include <smbsrv/libsmb.h>
#include <smbsrv/libmlrpc.h>
#include <smbsrv/smb_sid.h>

/*
 * Allocate a heap structure and the first heap block.  For many RPC
 * operations this will be the only time we need to malloc memory
 * in this instance of the heap.  The only point of note here is that
 * we put the heap management data in the first block to avoid a
```



```
 * second malloc. Make sure that sizeof(ndr_heap_t) is smaller
 * than NDR_HEAP_BLKSZ.
 *
 * Note that the heap management data is at the start of the first block.
 *
 * Returns a pointer to the newly created heap, which is used like an
 * opaque handle with the rest of the heap management interface..
 */
ndr_heap_t *
ndr_heap_create(void)
{
        ndr_heap_t *heap;
        char *base;
        size_t allocsize = sizeof (ndr_heap_t) + NDR_HEAP_BLKSZ;

        if ((heap = malloc(allocsize)) == NULL)
                return (NULL);

        base = (char *)heap;
        bzero(heap, sizeof (ndr_heap_t));

        heap->iovcnt = NDR_HEAP_MAXIOV;
        heap->iov = heap->iovec;
        heap->iov->iov_base = base;
        heap->iov->iov_len = sizeof (ndr_heap_t);
        heap->top = base + allocsize;
        heap->next = base + sizeof (ndr_heap_t);

        return (heap);
}
```

*_____unchanged_portion_omitted_*

```
/*
 * Convenience function to copy some memory into the heap.
 */
void *
ndr_heap_dupmem(ndr_heap_t *heap, const void *mem, size_t len)
{
        void *p;

        if (mem == NULL)
                return (NULL);

        if ((p = ndr_heap_malloc(heap, len)) != NULL)
                (void) memcpy(p, mem, len);

        return (p);
}

/*
 * Convenience function to do heap strdup.
 */
void *
ndr_heap_strdup(ndr_heap_t *heap, const char *s)
{
        int len;
        void *p;

        if (s == NULL)
                return (NULL);

        /*
         * We don't need to clutter the heap with empty strings.
         */
        if ((len = strlen(s)) == 0)
                return ("");
```

```
192         p = ndr_heap_dupmem(heap, s, len+1);
174         if ((p = ndr_heap_malloc(heap, len+1)) != NULL)
175                 (void) strcpy((char *)p, s);

194         return (p);
195 }

197 /*
198  * Make an ndr_mstring_t from a regular string.
199  */
200 int
201 ndr_heap_mstring(ndr_heap_t *heap, const char *s, ndr_mstring_t *out)
202 {
203         size_t slen;

205         if (s == NULL || out == NULL)
206                 return (-1);

208         /*
209          * Determine the WC strlen of s
210          * Was ndr__wcequiv_strlen(s)
211          */
212         slen = ndr__mbstowcs(NULL, s, NDR_STRING_MAX);
213         if (slen == (size_t)-1)
214                 return (-1);
189         out->length = smb_wcequiv_strlen(s);
190         out->allosize = out->length + sizeof (smb_wchar_t);

216         out->length = slen * sizeof (ndr_wchar_t);
217         out->allosize = out->length + sizeof (ndr_wchar_t);

219         if ((out->str = ndr_heap_strdup(heap, s)) == NULL)
220                 return (-1);

222         return (0);
223 }

225 /*
226  * Our regular string marshalling always creates null terminated strings
227  * but some Windows clients and servers are pedantic about the string
228  * formats they will accept and require non-null terminated strings.
229  * This function can be used to build a wide-char, non-null terminated
230  * string in the heap as a varying/conformant array.  We need to do the
231  * wide-char conversion here because the marshalling code won't be
232  * aware that this is really a string.
233  */
234 void
235 ndr_heap_mkvcs(ndr_heap_t *heap, char *s, ndr_vcstr_t *vc)
236 {
237         size_t slen;
238         int mlen;

240         /*
241          * Determine the WC strlen of s
242          * Was ndr__wcequiv_strlen(s)
243          */
244         slen = ndr__mbstowcs(NULL, s, NDR_STRING_MAX);
245         if (slen == (size_t)-1)
246                 slen = 0;

248         vc->wclen = slen * sizeof (ndr_wchar_t);
212         vc->wclen = smb_wcequiv_strlen(s);
249         vc->wcsize = vc->wclen;

251         /*
```



```
252          * alloc one extra wchar for a null
253          * See slen + 1 arg for mbstowcs
254          */
255         mlen = sizeof (ndr_vcs_t) + vc->wcsize + sizeof (ndr_wchar_t);
215         mlen = sizeof (ndr_vcs_t) + vc->wcsize + sizeof (smb_wchar_t);

256         vc->vcs = ndr_heap_malloc(heap, mlen);

258         if (vc->vcs) {
259                 vc->vcs->vc_first_is = 0;
260                 vc->vcs->vc_length_is = slen;
261                 (void) ndr__mbstowcs(vc->vcs->buffer, s, slen + 1);
221                 vc->vcs->vc_length_is = vc->wclen / sizeof (smb_wchar_t);
222                 (void) smb_mbstowcs((smb_wchar_t *)vc->vcs->buffer, s,
223                     vc->vcs->vc_length_is);
262         }
263 }
______unchanged_portion_omitted_

290 /*
291  * Removed ndr_heap_siddup(), now using ndr_heap_dupmem().
253  * Duplcate a SID in the heap.
292  */
255 smb_sid_t *
256 ndr_heap_siddup(ndr_heap_t *heap, smb_sid_t *sid)
257 {
258         smb_sid_t *new_sid;
259         unsigned size;

261         if (sid == NULL)
262                 return (NULL);

264         size = smb_sid_len(sid);

266         if ((new_sid = ndr_heap_malloc(heap, size)) == NULL)
267                 return (NULL);

269         bcopy(sid, new_sid, size);
270         return (new_sid);
271 }

294 int
295 ndr_heap_used(ndr_heap_t *heap)
296 {
297         int used = 0;
298         int i;

300         for (i = 0; i < NDR_HEAP_MAXIOV; ++i)
301                 used += heap->iovec[i].iov_len;

303         return (used);
304 }
______unchanged_portion_omitted_
```

****
15908 Sun Mar 18 01:12:58 2018
new/usr/src/lib/libmlrpc/common/ndr_marshal.c
1575 untangle libmlrpc ... (libmlrpc)
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 2007, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 */

#include <assert.h>
#include <strings.h>
#include <sys/param.h>

#include <libmlrpc.h>
#include <smbsrv/libsmb.h>
#include <smbsrv/libmlrpc.h>

#ifdef _BIG_ENDIAN
static const int ndr_native_byte_order = NDR_REPLAB_INTG_BIG_ENDIAN;
#else
static const int ndr_native_byte_order = NDR_REPLAB_INTG_LITTLE_ENDIAN;
#endif

static int ndr_decode_hdr_common(ndr_stream_t *, ndr_common_header_t *);
static int ndr_decode_pac_hdr(ndr_stream_t *, ndr_pac_hdr_t *);

static int
ndr_encode_decode_common(ndr_stream_t *nds, unsigned opnum,
    ndr_typeinfo_t *ti, void *datum)
{
        int rc;

        /*
         * Perform the (un)marshalling
         */
        if (ndo_operation(nds, ti, opnum, datum))
                return (NDR_DRC_OK);

        switch (nds->error) {
        case NDR_ERR_MALLOC_FAILED:
                rc = NDR_DRC_FAULT_OUT_OF_MEMORY;
                break;

        case NDR_ERR_SWITCH_VALUE_INVALID:
                rc = NDR_DRC_FAULT_PARAM_0_INVALID;
```



```
                break;

        case NDR_ERR_UNDERFLOW:
                rc = NDR_DRC_FAULT_RECEIVED_RUNT;
                break;

        case NDR_ERR_GROW_FAILED:
                rc = NDR_DRC_FAULT_ENCODE_TOO_BIG;
                break;

        default:
                if (nds->m_op == NDR_M_OP_MARSHALL)
                        rc = NDR_DRC_FAULT_ENCODE_FAILED;
                else
                        rc = NDR_DRC_FAULT_DECODE_FAILED;
                break;
        }

        return (rc);
}
```

*_____unchanged_portion_omitted_*

****
12982 Sun Mar 18 01:12:58 2018
new/usr/src/lib/libmlrpc/common/ndr_ops.c
1575 untangle libmlrpc ... (libmlrpc)
****

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

/*
 * Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 */

/*
 * Server-side NDR stream (PDU) operations. Stream operations should
 * return TRUE (non-zero) on success or FALSE (zero or a null pointer)
 * on failure. When an operation returns FALSE, including ndo_malloc()
 * returning NULL, it should set the nds->error to indicate what went
 * wrong.
 *
 * When available, the relevant ndr reference is passed to the
 * operation but keep in mind that it may be a null pointer.
 *
 * Functions ndo_get_pdu(), ndo_put_pdu(), and ndo_pad_pdu()
 * must never grow the PDU data. A request for out-of-bounds data is
 * an error. The swap_bytes flag is 1 if NDR knows that the byte-
 * order in the PDU is different from the local system.
 */

#include <sys/types.h>
#include <stdarg.h>
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <string.h>
#include <assert.h>

#include <libmlrpc.h>
#include <smbsrv/libsmb.h>
#include <smbsrv/libmlrpc.h>

#define NDOBUFSZ                128

#define NDR_PDU_BLOCK_SIZE      (4*1024)
#define NDR_PDU_BLOCK_MASK      (NDR_PDU_BLOCK_SIZE - 1)
#define NDR_PDU_ALIGN(N) \
        (((N) + NDR_PDU_BLOCK_SIZE) & ~NDR_PDU_BLOCK_MASK)
```



```
#define NDR_PDU_MAX_SIZE                (64*1024*1024)

static char *ndo_malloc(ndr_stream_t *, unsigned, ndr_ref_t *);
static int ndo_free(ndr_stream_t *, char *, ndr_ref_t *);
static int ndo_grow_pdu(ndr_stream_t *, unsigned long, ndr_ref_t *);
static int ndo_pad_pdu(ndr_stream_t *, unsigned long, unsigned long,
    ndr_ref_t *);
static int ndo_get_pdu(ndr_stream_t *, unsigned long, unsigned long,
    char *, int, ndr_ref_t *);
static int ndo_put_pdu(ndr_stream_t *, unsigned long, unsigned long,
    char *, int, ndr_ref_t *);
static void ndo_tattle(ndr_stream_t *, char *, ndr_ref_t *);
static void ndo_tattle_error(ndr_stream_t *, ndr_ref_t *);
static int ndo_reset(ndr_stream_t *);
static void ndo_destruct(ndr_stream_t *);
static void ndo_hexfmt(uint8_t *, int, int, char *, int);

/*
 * The ndr stream operations table.
 */
static ndr_stream_ops_t nds_ops = {
    ndo_malloc,
    ndo_free,
    ndo_grow_pdu,
    ndo_pad_pdu,
    ndo_get_pdu,
    ndo_put_pdu,
    ndo_tattle,
    ndo_tattle_error,
    ndo_reset,
    ndo_destruct
};
```

_____unchanged_portion_omitted_

```
**********************************************************
   54673 Sun Mar 18 01:12:59 2018
new/usr/src/lib/libmlrpc/common/ndr_process.c
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 * Copyright 2012 Milan Jurik. All rights reserved.
 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 */

/*
 * Network Data Representation (NDR) is a compatible subset of the DCE RPC
 * and MSRPC NDR.  NDR is used to move parameters consisting of
 * complicated trees of data constructs between an RPC client and server.
 */

#include <sys/byteorder.h>
#include <strings.h>
#include <assert.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

#include <libmlrpc.h>
#include <ndr_wchar.h>
#include <smbsrv/libsmb.h>
#include <smbsrv/string.h>
#include <smbsrv/libmlrpc.h>

#define NDR_STRING_MAX          4096

#define NDR_IS_UNION(T) \
        (((T)->type_flags & NDR_F_TYPEOP_MASK) == NDR_F_UNION)
#define NDR_IS_STRING(T)        \
        (((T)->type_flags & NDR_F_TYPEOP_MASK) == NDR_F_STRING)

extern ndr_typeinfo_t ndt_s_wchar;

/*
 * The following synopsis describes the terms TOP-MOST, OUTER and INNER.
 *
 * Each parameter (call arguments and return values) is a TOP-MOST item.
 * A TOP-MOST item consists of one or more OUTER items.  An OUTER item
 * consists of one or more INNER items.  There are important differences
```



```
 * between each kind, which, primarily, have to do with the allocation
 * of memory to contain data structures and the order of processing.
 *
 * This is most easily demonstrated with a short example.
 * Consider these structures:
 *
 *      struct top_param {
 *              long            level;
 *              struct list *   head;
 *              long            count;
 *      };
 *
 *      struct list {
 *              struct list *   next;
 *              char *          str; // a string
 *      };
 *
 * Now, consider an instance tree like this:
 *
 *      +---------+       +-------+       +-------+
 *      |top_param|  +--->|list #1|  +--->|list #2|
 *      +---------+  |    +-------+  |    +-------+
 *      | level   |  |    | next ----+    | next --->(NULL)
 *      | head   ----+    | str  -->"foo" | str  -->"bar"
 *      | count   |       | flag  |       | flag  |
 *      +---------+       +-------+       +-------+
 *
 * The DCE(MS)/RPC Stub Data encoding for the tree is the following.
 * The vertical bars (|) indicate OUTER construct boundaries.
 *
 *   +-----+----------------------+----------------------+-----+-----+-----+
 *   |level|#1.next #1.str #1.flag|#2.next #2.str #2.flag|"bar"|"foo"|count|
 *   +-----+----------------------+----------------------+-----+-----+-----+
 *   level |<---------------------- head -------------------------->|count
 *   TOP    TOP                                                     TOP
 *
 * Here's what to notice:
 *
 * - The members of the TOP-MOST construct are scattered through the Stub
 *   Data in the order they occur.  This example shows a TOP-MOST construct
 *   consisting of atomic types (pointers and integers).  A construct
 *   (struct) within the TOP-MOST construct would be contiguous and not
 *   scattered.
 *
 * - The members of OUTER constructs are contiguous, which allows for
 *   non-copied relocated (fixed-up) data structures at the packet's
 *   destination.  We don't do fix-ups here.  The pointers within the
 *   OUTER constructs are processed depth-first in the order that they
 *   occur.  If they were processed breadth first, the sequence would
 *   be #1,"foo",#2,"bar".  This is tricky because OUTER constructs may
 *   be variable length, and pointers are often encountered before the
 *   size(s) is known.
 *
 * - The INNER constructs are simply the members of an OUTER construct.
 *
 * For comparison, consider how ONC RPC would handle the same tree of
 * data.  ONC requires very little buffering, while DCE requires enough
 * buffer space for the entire message.  ONC does atom-by-atom depth-first
 * (de)serialization and copy, while DCE allows for constructs to be
 * "fixed-up" (relocated) in place at the destination.  The packet data
 * for the same tree processed by ONC RPC would look like this:
 *
 *   +-------------------------------------------------------------------+
 *   |level #1.next #2.next #2.str "bar" #2.flag #1.str "foo" #1.flag count|
 *   +-------------------------------------------------------------------+
 *   TOP    #1      #2      #2     bar   #2      #1     foo   #1      TOP
```

```
 123  *
 124  * More details about each TOP-MOST, OUTER, and INNER constructs appear
 125  * throughout this source file near where such constructs are processed.
 126  *
 127  * NDR_REFERENCE
 128  *
 129  * The primary object for NDR is the ndr_ref_t.
 130  *
 131  * An ndr reference indicates the local datum (i.e. native "C" data
 132  * format), and the element within the Stub Data (contained within the
 133  * RPC PDU (protocol data unit).  An ndr reference also indicates,
 134  * largely as a debugging aid, something about the type of the
 135  * element/datum, and the enclosing construct for the element. The
 136  * ndr reference's are typically allocated on the stack as locals,
 137  * and the chain of ndr-reference.enclosing references is in reverse
 138  * order of the call graph.
 139  *
 140  * The ndr-reference.datum is a pointer to the local memory that
 141  * contains/receives the value. The ndr-reference.pdu_offset indicates
 142  * where in the Stub Data the value is to be stored/retrieved.
 143  *
 144  * The ndr-reference also contains various parameters to the NDR
 145  * process, such as ndr-reference.size_is, which indicates the size
 146  * of variable length data, or ndr-reference.switch_is, which
 147  * indicates the arm of a union to use.
 148  *
 149  * QUEUE OF OUTER REFERENCES
 150  *
 151  * Some OUTER constructs are variable size.  Sometimes (often) we don't
 152  * know the size of the OUTER construct until after pointers have been
 153  * encountered. Hence, we can not begin processing the referent of the
 154  * pointer until after the referring OUTER construct is completely
 155  * processed, i.e. we don't know where to find/put the referent in the
 156  * Stub Data until we know the size of all its predecessors.
 157  *
 158  * This is managed using the queue of OUTER references.  The queue is
 159  * anchored in ndr_stream.outer_queue_head.  At any time,
 160  * ndr_stream.outer_queue_tailp indicates where to put the
 161  * ndr-reference for the next encountered pointer.
 162  *
 163  * Refer to the example above as we illustrate the queue here.  In these
 164  * illustrations, the queue entries are not the data structures themselves.
 165  * Rather, they are ndr-reference entries which **refer** to the data
 166  * structures in both the PDU and local memory.
 167  *
 168  * During some point in the processing, the queue looks like this:
 169  *
 170  *   outer_current -------v
 171  *   outer_queue_head --> list#1 --0
 172  *   outer_queue_tailp ---------&
 173  *
 174  * When the pointer #1.next is encountered, and entry is added to the
 175  * queue,
 176  *
 177  *   outer_current -------v
 178  *   outer_queue_head --> list#1 --> list#2 --0
 179  *   outer_queue_tailp --------------------&
 180  *
 181  * and the members of #1 continue to be processed, which encounters
 182  * #1.str:
 183  *
 184  *   outer_current -------v
 185  *   outer_queue_head --> list#1 --> list#2 --> "foo" --0
 186  *   outer_queue_tailp -----------------------------&
 187  *
 188  * Upon the completion of list#1, the processing continues by moving to
```



```
 189  * ndr_stream.outer_current->next, and the tail is set to this outer member:
 190  *
 191  *   outer_current ------------------v
 192  *   outer_queue_head --> list#1 --> list#2 --> "foo" --0
 193  *   outer_queue_tailp --------------------&
 194  *
 195  * Space for list#2 is allocated, either in the Stub Data or of local
 196  * memory.  When #2.next is encountered, it is found to be the null
 197  * pointer and no reference is added to the queue.  When #2.str is
 198  * encountered, it is found to be valid, and a reference is added:
 199  *
 200  *   outer_current ------------------v
 201  *   outer_queue_head --> list#1 --> list#2 --> "bar" --> "foo" --0
 202  *   outer_queue_tailp -----------------------------&
 203  *
 204  * Processing continues in a similar fashion with the string "bar",
 205  * which is variable-length.  At this point, memory for "bar" may be
 206  * malloc()ed during NDR_M_OP_UNMARSHALL:
 207  *
 208  *   outer_current -----------------------------v
 209  *   outer_queue_head --> list#1 --> list#2 --> "bar" --> "foo" --0
 210  *   outer_queue_tailp -----------------------------&
 211  *
 212  * And finishes on string "foo".  Notice that because "bar" is a
 213  * variable length string, and we don't know the PDU offset for "foo"
 214  * until we reach this point.
 215  *
 216  * When the queue is drained (current->next==0), processing continues
 217  * with the next TOP-MOST member.
 218  *
 219  * The queue of OUTER constructs manages the variable-length semantics
 220  * of OUTER constructs and satisfies the depth-first requirement.
 221  * We allow the queue to linger until the entire TOP-MOST structure is
 222  * processed as an aid to debugging.
 223  */

 225 static ndr_ref_t *ndr_enter_outer_queue(ndr_ref_t *);
 226 extern int ndr__ulong(ndr_ref_t *);

 228 /*
 229  * TOP-MOST ELEMENTS
 230  *
 231  * This is fundamentally the first OUTER construct of the parameter,
 232  * possibly followed by more OUTER constructs due to pointers.  The
 233  * datum (local memory) for TOP-MOST constructs (structs) is allocated
 234  * by the caller of NDR.
 235  *
 236  * After the element is transferred, the outer_queue is drained.
 237  *
 238  * All we have to do is add an entry to the outer_queue for this
 239  * top-most member, and commence the outer_queue processing.
 240  */
 241 int
 242 ndo_process(ndr_stream_t *nds, ndr_typeinfo_t *ti, char *datum)
 243 {
 244         ndr_ref_t       myref;

 246         bzero(&myref, sizeof (myref));
 247         myref.stream = nds;
 248         myref.datum = datum;
 249         myref.name = "PROCESS";
 250         myref.ti = ti;

 252         return (ndr_topmost(&myref));
 253 }
```

_____***unchanged_portion_omitted_***

```
1160 int
1161 ndr_outer_string(ndr_ref_t *outer_ref)
1162 {
1163         ndr_stream_t    *nds = outer_ref->stream;
1164         ndr_typeinfo_t  *ti = outer_ref->ti;
1165         ndr_ref_t       myref;
1166         char            *valp = NULL;
1167         unsigned        is_varlen = ti->pdu_size_variable_part;
1168         int             is_union = NDR_IS_UNION(ti);
1169         int             is_string = NDR_IS_STRING(ti);
1170         int             rc;
1171         unsigned        n_zeroes;
1172         unsigned        ix;
1173         unsigned long   size_is;
1174         unsigned long   first_is;
1175         unsigned long   length_is;
1176         unsigned        n_hdr;
1177         unsigned        n_fixed;
1178         unsigned        n_variable;
1179         unsigned        n_alloc;
1180         unsigned        n_pdu_total;
1181         int             params;

1183         params = outer_ref->outer_flags & NDR_F_PARAMS_MASK;

1185         assert(is_varlen && is_string && !is_union);
1186         assert(params == NDR_F_NONE);

1188         /* string header for this: size_is first_is length_is */
1189         n_hdr = 12;

1191         /* fixed part -- exactly none of these */
1192         n_fixed = 0;

1194         if (!ndr_outer_grow(outer_ref, n_hdr))
1195                 return (0);             /* error already set */

1197         switch (nds->m_op) {
1198         case NDR_M_OP_MARSHALL:
1199                 valp = outer_ref->datum;
1200                 if (!valp) {
1201                         NDR_SET_ERROR(outer_ref, NDR_ERR_OUTER_PARAMS_BAD);
1202                         return (0);
1203                 }

1205                 if (outer_ref->backptr)
1206                         assert(valp == *outer_ref->backptr);

1208                 if (ti == &ndt_s_wchar) {
1209                         /*
1210                          * size_is is the number of characters in the
1211                          * (multibyte) string, including the null.
1212                          * In other words, symbols, not bytes.
1213                          */
1214                         size_t wlen;
1215                         wlen = ndr__mbstowcs(NULL, valp, NDR_STRING_MAX);
1216                         if (wlen == (size_t)-1) {
1217                                 /* illegal sequence error? */
1214                         size_is = smb_wcequiv_strlen(valp) /
1215                             sizeof (smb_wchar_t);

1217                         if (!(nds->flags & NDS_F_NONULL))
1218                                 ++size_is;

1220                         if (size_is > NDR_STRING_MAX) {
```



```
1218                                 NDR_SET_ERROR(outer_ref, NDR_ERR_STRLEN);
1219                                 return (0);
1220                         }
1221                         if ((nds->flags & NDS_F_NONULL) == 0)
1222                                 wlen++;
1223                         if (wlen > NDR_STRING_MAX) {
1224                                 NDR_SET_ERROR(outer_ref, NDR_ERR_STRLEN);
1225                                 return (0);
1226                         }
1227                         size_is = wlen;
1228                 } else {
1229                         valp = outer_ref->datum;
1230                         n_zeroes = 0;
1231                         for (ix = 0; ix < NDR_STRING_MAX; ix++) {
1232                                 if (valp[ix] == 0) {
1233                                         n_zeroes++;
1234                                         if (n_zeroes >= is_varlen &&
1235                                             ix % is_varlen == 0) {
1236                                                 break;
1237                                         }
1238                                 } else {
1239                                         n_zeroes = 0;
1240                                 }
1241                         }
1242                         if (ix >= NDR_STRING_MAX) {
1243                                 NDR_SET_ERROR(outer_ref, NDR_ERR_STRLEN);
1244                                 return (0);
1245                         }
1246                         size_is = ix+1;
1247                 }

1249                 first_is = 0;

1251                 if (nds->flags & NDS_F_NOTERM)
1252                         length_is = size_is - 1;
1253                 else
1254                         length_is = size_is;

1256                 if (!ndr_outer_poke_sizing(outer_ref, 0, &size_is) ||
1257                     !ndr_outer_poke_sizing(outer_ref, 4, &first_is) ||
1258                     !ndr_outer_poke_sizing(outer_ref, 8, &length_is))
1259                         return (0);             /* error already set */
1260                 break;

1262         case NDR_M_OP_UNMARSHALL:
1263                 if (!ndr_outer_peek_sizing(outer_ref, 0, &size_is) ||
1264                     !ndr_outer_peek_sizing(outer_ref, 4, &first_is) ||
1265                     !ndr_outer_peek_sizing(outer_ref, 8, &length_is))
1266                         return (0);             /* error already set */

1268                 /*
1269                  * In addition to the first_is check, we used to check that
1270                  * size_is or size_is-1 was equal to length_is but Windows95
1271                  * doesn't conform to this "rule" (see variable part below).
1272                  * The srvmgr tool for Windows95 sent the following values
1273                  * for a path string:
1274                  *
1275                  *      size_is   = 261 (0x105)
1276                  *      first_is  = 0
1277                  *      length_is = 53  (0x35)
1278                  *
1279                  * The length_is was correct (for the given path) but the
1280                  * size_is was the maximum path length rather than being
1281                  * related to length_is.
1282                  */
1283                 if (first_is != 0) {
```

```
1284                        NDR_SET_ERROR(outer_ref, NDR_ERR_STRING_SIZING);
1285                        return (0);
1286                }

1288                if (ti == &ndt_s_wchar) {
1289                        /*
1290                         * Decoding Unicode to UTF-8; we need to allow
1291                         * for the maximum possible char size. It would
1292                         * be nice to use mbequiv_strlen but the string
1293                         * may not be null terminated.
1294                         */
1295                        n_alloc = (size_is + 1) * NDR_MB_CHAR_MAX;
1291                        n_alloc = (size_is + 1) * MTS_MB_CHAR_MAX;
1296                } else {
1297                        n_alloc = (size_is + 1) * is_varlen;
1298                }

1300                valp = NDS_MALLOC(nds, n_alloc, outer_ref);
1301                if (!valp) {
1302                        NDR_SET_ERROR(outer_ref, NDR_ERR_MALLOC_FAILED);
1303                        return (0);
1304                }

1306                bzero(valp, (size_is+1) * is_varlen);

1308                if (outer_ref->backptr)
1309                        *outer_ref->backptr = valp;
1310                outer_ref->datum = valp;
1311                break;

1313        default:
1314                NDR_SET_ERROR(outer_ref, NDR_ERR_M_OP_INVALID);
1315                return (0);
1316        }

1318        /*
1319         * Variable part - exactly length_is of these.
1320         *
1321         * Usually, length_is is same as size_is and includes nul.
1322         * Some protocols use length_is = size_is-1, and length_is does
1323         * not include the nul (which is more consistent with DCE spec).
1324         * If the length_is is 0, there is no data following the
1325         * sizing header, regardless of size_is.
1326         */
1327        n_variable = length_is * is_varlen;

1329        /* sum them up to determine the PDU space required */
1330        n_pdu_total = n_hdr + n_fixed + n_variable;

1332        /* similar sum to determine how much local memory is required */
1333        n_alloc = n_fixed + n_variable;

1335        rc = ndr_outer_grow(outer_ref, n_pdu_total);
1336        if (!rc)
1337                return (rc);            /* error already set */

1339        if (length_is > 0) {
1340                bzero(&myref, sizeof (myref));
1341                myref.stream = nds;
1342                myref.enclosing = outer_ref;
1343                myref.ti = outer_ref->ti;
1344                myref.datum = outer_ref->datum;
1345                myref.name = "OUTER-STRING";
1346                myref.outer_flags = NDR_F_IS_STRING;
1347                myref.inner_flags = NDR_F_NONE;
```



```
1349                /*
1350                 * Set up size_is and strlen_is for ndr_s_wchar.
1351                 */
1352                myref.size_is = size_is;
1353                myref.strlen_is = length_is;
1354        }

1356        myref.pdu_offset = outer_ref->pdu_offset + 12;

1358        /*
1359         * Don't try to decode empty strings.
1360         */
1361        if ((size_is == 0) && (first_is == 0) && (length_is == 0)) {
1362                nds->pdu_scan_offset = outer_ref->pdu_end_offset;
1363                return (1);
1364        }

1366        if ((size_is != 0) && (length_is != 0)) {
1367                rc = ndr_inner(&myref);
1368                if (!rc)
1369                        return (rc);            /* error already set */
1370        }

1372        nds->pdu_scan_offset = outer_ref->pdu_end_offset;
1373        return (1);
1374 }
```

*unchanged_portion_omitted_*

```
1720 int
1721 ndr_inner_array(ndr_ref_t *encl_ref)
1722 {
1723        ndr_typeinfo_t          *ti = encl_ref->ti;
1724        ndr_ref_t               myref;
1725        unsigned long           pdu_offset = encl_ref->pdu_offset;
1726        unsigned long           n_elem;
1727        unsigned long           i;
1728        char                    name[30];

1730        if (encl_ref->inner_flags & NDR_F_SIZE_IS) {
1731                /* now is the time to check/set size */
1732                if (!ndr_size_is(encl_ref))
1733                        return (0);     /* error already set */
1734                n_elem = encl_ref->size_is;
1735        } else {
1736                assert(encl_ref->inner_flags & NDR_F_DIMENSION_IS);
1737                n_elem = encl_ref->dimension_is;
1738        }

1740        bzero(&myref, sizeof (myref));
1741        myref.enclosing = encl_ref;
1742        myref.stream = encl_ref->stream;
1743        myref.packed_alignment = 0;
1744        myref.ti = ti;
1745        myref.inner_flags = NDR_F_NONE;

1747        for (i = 0; i < n_elem; i++) {
1748                (void) snprintf(name, sizeof (name), "[%lu]", i);
1744                (void) sprintf(name, "[%lu]", i);
1749                myref.name = name;
1750                myref.pdu_offset = pdu_offset + i * ti->pdu_size_fixed_part;
1751                myref.datum = encl_ref->datum + i * ti->c_size_fixed_part;

1753                if (!ndr_inner(&myref))
1754                        return (0);
1755        }
```

```
1757         return (1);
1758 }
______unchanged_portion_omitted_

1796 #define MAKE_BASIC_TYPE(TYPE, SIZE) \
1797         MAKE_BASIC_TYPE_BASE(TYPE, SIZE) \
1798         MAKE_BASIC_TYPE_STRING(TYPE, SIZE)

1800 int ndr_basic_integer(ndr_ref_t *, unsigned);
1801 int ndr_string_basic_integer(ndr_ref_t *, ndr_typeinfo_t *);

1803 /* Comments to be nice to those searching for these types. */
1804 MAKE_BASIC_TYPE(_char, 1)       /* ndt__char,  ndt_s_char */
1805 MAKE_BASIC_TYPE(_uchar, 1)      /* ndt__uchar, ndt_s_uchar */
1806 MAKE_BASIC_TYPE(_short, 2)      /* ndt__short, ndt_s_short */
1807 MAKE_BASIC_TYPE(_ushort, 2)     /* ndt__ushort, ndt_s_ushort */
1808 MAKE_BASIC_TYPE(_long, 4)       /* ndt__long,  ndt_s_long */
1809 MAKE_BASIC_TYPE(_ulong, 4)      /* ndt__ulong, ndt_s_ulong */

1811 MAKE_BASIC_TYPE_BASE(_wchar, 2) /* ndt__wchar, ndt_s_wchar */
1800 MAKE_BASIC_TYPE(_char, 1)
1801 MAKE_BASIC_TYPE(_uchar, 1)
1802 MAKE_BASIC_TYPE(_short, 2)
1803 MAKE_BASIC_TYPE(_ushort, 2)
1804 MAKE_BASIC_TYPE(_long, 4)
1805 MAKE_BASIC_TYPE(_ulong, 4)

1807 MAKE_BASIC_TYPE_BASE(_wchar, 2)

1813 int
1814 ndr_basic_integer(ndr_ref_t *ref, unsigned size)
1815 {
1816         ndr_stream_t    *nds = ref->stream;
1817         char            *valp = (char *)ref->datum;
1818         int             rc;

1820         switch (nds->m_op) {
1821         case NDR_M_OP_MARSHALL:
1822                 rc = NDS_PUT_PDU(nds, ref->pdu_offset, size,
1823                     valp, nds->swap, ref);
1824                 break;

1826         case NDR_M_OP_UNMARSHALL:
1827                 rc = NDS_GET_PDU(nds, ref->pdu_offset, size,
1828                     valp, nds->swap, ref);
1829                 break;

1831         default:
1832                 NDR_SET_ERROR(ref, NDR_ERR_M_OP_INVALID);
1833                 return (0);
1834         }

1836         return (rc);
1837 }

1839 int
1840 ndr_string_basic_integer(ndr_ref_t *encl_ref, ndr_typeinfo_t *type_under)
1841 {
1842         unsigned long           pdu_offset = encl_ref->pdu_offset;
1843         unsigned                size = type_under->pdu_size_fixed_part;
1844         char                    *valp;
1845         ndr_ref_t               myref;
1846         unsigned long           i;
1847         long                    sense = 0;
1848         char                    name[30];
```



```
1850         assert(size != 0);

1852         bzero(&myref, sizeof (myref));
1853         myref.enclosing = encl_ref;
1854         myref.stream = encl_ref->stream;
1855         myref.packed_alignment = 0;
1856         myref.ti = type_under;
1857         myref.inner_flags = NDR_F_NONE;
1858         myref.name = name;

1860         for (i = 0; i < NDR_STRING_MAX; i++) {
1861                 (void) snprintf(name, sizeof (name), "[%lu]", i);
1857                 (void) sprintf(name, "[%lu]", i);
1862                 myref.pdu_offset = pdu_offset + i * size;
1863                 valp = encl_ref->datum + i * size;
1864                 myref.datum = valp;

1866                 if (!ndr_inner(&myref))
1867                         return (0);

1869                 switch (size) {
1870                 case 1:         sense = *valp; break;
1871                 /*LINTED E_BAD_PTR_CAST_ALIGN*/
1872                 case 2:         sense = *(short *)valp; break;
1873                 /*LINTED E_BAD_PTR_CAST_ALIGN*/
1874                 case 4:         sense = *(long *)valp; break;
1875                 }

1877                 if (!sense)
1878                         break;
1879         }

1881         return (1);
1882 }
______unchanged_portion_omitted_


1898 /*
1899  * Hand coded wchar function because all strings are transported
1900  * as wide characters. During NDR_M_OP_MARSHALL, we convert from
1901  * multi-byte to wide characters. During NDR_M_OP_UNMARSHALL, we
1902  * convert from wide characters to multi-byte.
1903  *
1904  * The most critical thing to get right in this function is to
1905  * marshall or unmarshall _exactly_ the number of elements the
1906  * OtW length specifies, as saved by the caller in: strlen_is.
1907  * Doing otherwise would leave us positioned at the wrong place
1908  * in the data stream for whatever follows this.  Note that the
1909  * string data covered by strlen_is may or may not include any
1910  * null termination, but the converted string provided by the
1911  * caller or returned always has a null terminator.
1900  * It appeared that NT would sometimes leave a spurious character
1901  * in the data stream before the null wide_char, which would get
1902  * included in the string decode because we processed until the
1903  * null character. It now looks like NT does not always terminate
1904  * RPC Unicode strings and the terminating null is a side effect
1905  * of field alignment. So now we rely on the strlen_is (set up in
1906  * ndr_outer_string) of the enclosing reference. This may or may
1907  * not include the null but it doesn't matter, the algorithm will
1908  * get it right.
1912  */
1913 int
1914 ndr_s_wchar(ndr_ref_t *encl_ref)
1915 {
1916         ndr_stream_t            *nds = encl_ref->stream;
1917         char                    *valp = encl_ref->datum;
```

```
1914        unsigned short          wide_char;
1915        char                    *valp;
1918        ndr_ref_t               myref;
1917        unsigned long           i;
1919        char                    name[30];
1920        ndr_wchar_t             wcs[NDR_STRING_MAX+1];
1921        size_t                  i, slen, wlen;
1919        int                     count;
1920        int                     char_count = 0;

1923        /* This is enforced in ndr_outer_string() */
1924        assert(encl_ref->strlen_is <= NDR_STRING_MAX);

1926        if (nds->m_op == NDR_M_OP_UNMARSHALL) {
1927                /*
1928                 * To avoid problems with zero length strings
1929                 * we can just null terminate here and be done.
1930                 */
1931                if (encl_ref->strlen_is == 0) {
1932                        encl_ref->datum[0] = '\0';
1933                        return (1);
1934                }
1935        }

1937        /*
1938         * If we're marshalling, convert the given string
1939         * from UTF-8 into a local UCS-2 string.
1940         */
1941        if (nds->m_op == NDR_M_OP_MARSHALL) {
1942                wlen = ndr__mbstowcs(wcs, valp, NDR_STRING_MAX);
1943                if (wlen == (size_t)-1)
1944                        return (0);
1945                /*
1946                 * Add a nulls to make strlen_is.
1947                 * (always zero or one of them)
1948                 * Then null terminate at wlen,
1949                 * just for debug convenience.
1950                 */
1951                while (wlen < encl_ref->strlen_is)
1952                        wcs[wlen++] = 0;
1953                wcs[wlen] = 0;
1954        }

1956        /*
1957         * Copy wire data to or from the local wc string.
1958         * Always exactly strlen_is elements.
1959         */
1960        bzero(&myref, sizeof (myref));
1961        myref.enclosing = encl_ref;
1962        myref.stream = encl_ref->stream;
1963        myref.packed_alignment = 0;
1964        myref.ti = &ndt__wchar;
1965        myref.inner_flags = NDR_F_NONE;
1939        myref.datum = (char *)&wide_char;
1966        myref.name = name;
1967        myref.pdu_offset = encl_ref->pdu_offset;
1968        myref.datum = (char *)wcs;
1969        wlen = encl_ref->strlen_is;

1971        for (i = 0; i < wlen; i++) {
1972                (void) snprintf(name, sizeof (name), "[%lu]", i);
1973                if (!ndr_inner(&myref))
1943        valp = encl_ref->datum;
1944        count = 0;

1946        for (i = 0; i < NDR_STRING_MAX; i++) {
```



```
1947                (void) sprintf(name, "[%lu]", i);

1949                if (nds->m_op == NDR_M_OP_MARSHALL) {
1950                        count = smb_mbtowc((smb_wchar_t *)&wide_char, valp,
1951                            MTS_MB_CHAR_MAX);
1952                        if (count < 0) {
1974                        return (0);
1975                myref.pdu_offset += sizeof (ndr_wchar_t);
1976                myref.datum      += sizeof (ndr_wchar_t);
1977        }
1954                        } else if (count == 0) {
1955                                if (encl_ref->strlen_is != encl_ref->size_is)
1956                                        break;

1979        /*
1980         * If this is unmarshall, convert the local UCS-2 string
1981         * into a UTF-8 string in the caller's buffer.  The caller
1982         * previously determined the space required and provides a
1983         * buffer of sufficient size.
1959                                 * If the input char is 0, mbtowc
1960                                 * returns 0 without setting wide_char.
1961                                 * Set wide_char to 0 and a count of 1.
1984         */
1985        if (nds->m_op == NDR_M_OP_UNMARSHALL) {
1986                wcs[wlen] = 0;
1987                slen = ndr__wcstombs(valp, wcs, wlen);
1988                if (slen == (size_t)-1)
1963                                wide_char = *valp;
1964                                count = 1;
1965                        }
1966                }

1968                if (!ndr_inner(&myref))
1989                        return (0);
1990                valp[slen] = '\0';

1971                if (nds->m_op == NDR_M_OP_UNMARSHALL) {
1972                        count = smb_wctomb(valp, wide_char);

1974                        if ((++char_count) == encl_ref->strlen_is) {
1975                                valp += count;
1976                                *valp = '\0';
1977                                break;
1991        }
1979                }

1981                if (!wide_char)
1982                        break;

1984                myref.pdu_offset += sizeof (wide_char);
1985                valp += count;
1986        }

1993        return (1);
1994 }

1996 /*
1997  * Converts a multibyte character string to a little-endian, wide-char
1998  * string.  No more than nwchars wide characters are stored.
1999  * A terminating null wide character is appended if there is room.
2000  *
2001  * Returns the number of wide characters converted, not counting
2002  * any terminating null wide character.  Returns -1 if an invalid
2003  * multibyte character is encountered.
2004  */
2005 /* ARGSUSED */
```

```
2006 size_t
2007 ndr_mbstowcs(ndr_stream_t *nds, ndr_wchar_t *wcs, const char *mbs,
2001 ndr_mbstowcs(ndr_stream_t *nds, smb_wchar_t *wcs, const char *mbs,
2008     size_t nwchars)
2009 {
2010         size_t len;
2004         smb_wchar_t *start = wcs;
2005         int nbytes;

2007         while (nwchars--) {
2008                 nbytes = ndr_mbtowc(nds, wcs, mbs, MTS_MB_CHAR_MAX);
2009                 if (nbytes < 0) {
2010                         *wcs = 0;
2011                         return ((size_t)-1);
2012                 }

2014                 if (*mbs == 0)
2015                         break;

2017                 ++wcs;
2018                 mbs += nbytes;
2019         }

2021         return (wcs - start);
2022 }

2024 /*
2025  * Converts a multibyte character to a little-endian, wide-char, which
2026  * is stored in wcharp.  Up to nbytes bytes are examined.
2027  *
2028  * If mbchar is valid, returns the number of bytes processed in mbchar.
2029  * If mbchar is invalid, returns -1.  See also smb_mbtowc().
2030  */
2031 /*ARGSUSED*/
2032 int
2033 ndr_mbtowc(ndr_stream_t *nds, smb_wchar_t *wcharp, const char *mbchar,
2034     size_t nbytes)
2035 {
2036         int rc;

2038         if ((rc = smb_mbtowc(wcharp, mbchar, nbytes)) < 0)
2039                 return (rc);

2012 #ifdef _BIG_ENDIAN
2013         if (nds == NULL || NDR_MODE_MATCH(nds, NDR_MODE_RETURN_SEND)) {
2014                 /* Make WC string in LE order. */
2015                 len = ndr__mbstowcs_le(wcs, mbs, nwchars);
2016         } else
2042         if (nds == NULL || NDR_MODE_MATCH(nds, NDR_MODE_RETURN_SEND))
2043                 *wcharp = BSWAP_16(*wcharp);
2017 #endif
2018                 len = ndr__mbstowcs(wcs, mbs, nwchars);

2020         return (len);
2046         return (rc);
2021 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   20962 Sun Mar 18 01:12:59 2018
new/usr/src/lib/libmlrpc/common/ndr_server.c
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 2007, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright 2015 Nexenta Systems, Inc.  All rights reserved.
 */

/*
 * Server side RPC handler.
 */

#include <sys/byteorder.h>
#include <sys/uio.h>
#include <errno.h>
#include <synch.h>
#include <stdlib.h>
#include <strings.h>
#include <string.h>
#include <thread.h>

#include <libmlrpc.h>
#include <smbsrv/libsmb.h>
#include <smbsrv/libmlrpc.h>
#include <smbsrv/ntaccess.h>

#define NDR_PIPE_SEND(np, buf, len) \
        ((np)->np_send)((np), (buf), (len))
#define NDR_PIPE_RECV(np, buf, len) \
        ((np)->np_recv)((np), (buf), (len))

static int ndr_svc_process(ndr_xa_t *);
static int ndr_svc_bind(ndr_xa_t *);
static int ndr_svc_request(ndr_xa_t *);
static void ndr_reply_prepare_hdr(ndr_xa_t *);
static int ndr_svc_alter_context(ndr_xa_t *);
static void ndr_reply_fault(ndr_xa_t *, unsigned long);

static int ndr_recv_request(ndr_xa_t *mxa);
static int ndr_recv_frag(ndr_xa_t *mxa);
static int ndr_send_reply(ndr_xa_t *);

static int ndr_pipe_process(ndr_pipe_t *, ndr_xa_t *);
```



```
/*
 * External entry point called by smbd.
 */
void
ndr_pipe_worker(ndr_pipe_t *np)
{
        ndr_xa_t        *mxa;
        int rc;

        ndr_svc_binding_pool_init(&np->np_binding, np->np_binding_pool,
            NDR_N_BINDING_POOL);

        if ((mxa = malloc(sizeof (*mxa))) == NULL)
                return;

        do {
                bzero(mxa, sizeof (*mxa));
                rc = ndr_pipe_process(np, mxa);
        } while (rc == 0);

        free(mxa);

        /*
         * Ensure that there are no RPC service policy handles
         * (associated with this fid) left around.
         */
        ndr_hdclose(np);
}
```

_____*unchanged_portion_omitted_*

```
/*
 * Check whether or not the specified user has administrator privileges,
 * i.e. is a member of Domain Admins or Administrators.
 * Returns true if the user is an administrator, otherwise returns false.
 */
boolean_t
ndr_is_admin(ndr_xa_t *xa)
{
        smb_netuserinfo_t *ctx = xa->pipe->np_user;

        return (ctx->ui_flags & SMB_ATF_ADMIN);
}

/*
 * Check whether or not the specified user has power-user privileges,
 * i.e. is a member of Domain Admins, Administrators or Power Users.
 * This is typically required for operations such as managing shares.
 * Returns true if the user is a power user, otherwise returns false.
 */
boolean_t
ndr_is_poweruser(ndr_xa_t *xa)
{
        smb_netuserinfo_t *ctx = xa->pipe->np_user;

        return ((ctx->ui_flags & SMB_ATF_ADMIN) ||
            (ctx->ui_flags & SMB_ATF_POWERUSER));
}

int32_t
ndr_native_os(ndr_xa_t *xa)
{
        smb_netuserinfo_t *ctx = xa->pipe->np_user;

        return (ctx->ui_native_os);
}
```

```
132 /*
133  * Receive an entire RPC request (all fragments)
134  * Returns zero or an NDR fault code.
135  */
136 static int
137 ndr_recv_request(ndr_xa_t *mxa)
138 {
139         ndr_common_header_t     *hdr = &mxa->recv_hdr.common_hdr;
140         ndr_stream_t            *nds = &mxa->recv_nds;
141         unsigned long           saved_size;
142         int                     rc;

144         rc = ndr_recv_frag(mxa);
145         if (rc != 0)
146                 return (rc);
147         if (!NDR_IS_FIRST_FRAG(hdr->pfc_flags))
148                 return (NDR_DRC_FAULT_DECODE_FAILED);

150         while (!NDR_IS_LAST_FRAG(hdr->pfc_flags)) {
151                 rc = ndr_recv_frag(mxa);
152                 if (rc != 0)
153                         return (rc);
154         }
155         nds->pdu_scan_offset = 0;

157         /*
158          * This whacks nds->pdu_size, so save/restore.
159          * It leaves scan_offset after the header.
160          */
161         saved_size = nds->pdu_size;
162         rc = ndr_decode_pdu_hdr(mxa);
163         nds->pdu_size = saved_size;

165         return (rc);
166 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    9380 Sun Mar 18 01:12:59 2018
new/usr/src/lib/libmlrpc/common/ndr_svc.c
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 */

#include <uuid/uuid.h>
#include <ctype.h>
#include <synch.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <strings.h>
#include <assert.h>

#include <libmlrpc.h>
#include <smbsrv/libsmb.h>
#include <smbsrv/libmlrpc.h>


/*
 * Global list of allocated handles.  Handles are used in various
 * server-side RPC functions: typically, issued when a service is
 * opened and obsoleted when it is closed.  Clients should treat
 * handles as opaque data.
 */
static ndr_handle_t *ndr_handle_list;
static mutex_t ndr_handle_lock;

/*
 * Table of registered services.
 */
#define NDR_MAX_SERVICES        32
static ndr_service_t *ndr_services[NDR_MAX_SERVICES];

/*
 * Register a service.
 *
 * Returns:
 *      0       Success
```



```
 *      -1      Duplicate service
 *      -2      Duplicate name
 *      -3      Table overflow
 */
int
ndr_svc_register(ndr_service_t *svc)
{
        ndr_service_t   *p;
        int             free_slot = -1;
        int             i;

        for (i = 0; i < NDR_MAX_SERVICES; i++) {
                if ((p = ndr_services[i]) == NULL) {
                        if (free_slot < 0)
                                free_slot = i;
                        continue;
                }

                if (p == svc)
                        return (-1);

                if (strcasecmp(p->name, svc->name) == 0)
                        return (-2);
        }

        if (free_slot < 0)
                return (-3);

        ndr_services[free_slot] = svc;
        return (0);
}
_____unchanged_portion_omitted_
```

```
**********************************************************
    3720 Sun Mar 18 01:12:59 2018
new/usr/src/lib/libmlrpc/common/ndr_wchar.c
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

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

 22 /*
 23  * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 24  * Use is subject to license terms.
 25  *
 26  * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 27  */

 29 /*
 30  * Some wchar support functions used by this library.
 31  * Mostlly just wrappers that call sys/u8_textprep.h
 32  * functions: uconv_u8tou16, uconv_u16tou8.
 33  */

 35 #include <sys/types.h>
 36 #include <sys/u8_textprep.h>
 37 #include <string.h>

 39 #include "ndr_wchar.h"

 41 /*
 42  * When we just want lengths, we need an output buffer to pass to the
 43  * uconv_... functions.  Nothing ever reads this output, so we can
 44  * use shared space for the unwanted output.
 45  */
 46 static uint16_t junk_wcs[NDR_STRING_MAX];
 47 static char junk_mbs[NDR_MB_CUR_MAX * NDR_STRING_MAX];

 49 static size_t
 50 ndr__mbstowcs_x(uint16_t *, const char *, size_t, int);

 52 /*
 53  * Like mbstowcs(3C), but with UCS-2 wchar_t
 54  */
 55 size_t
 56 ndr__mbstowcs(uint16_t *wcs, const char *mbs, size_t nwchars)
 57 {
 58         return (ndr__mbstowcs_x(wcs, mbs, nwchars,
 59             UCONV_OUT_SYSTEM_ENDIAN));
 60 }
```



```
 62 /*
 63  * Like above, but put UCS-2 little-endian.
 64  */
 65 size_t
 66 ndr__mbstowcs_le(uint16_t *wcs, const char *mbs, size_t nwchars)
 67 {
 68         return (ndr__mbstowcs_x(wcs, mbs, nwchars,
 69             UCONV_OUT_LITTLE_ENDIAN));
 70 }

 72 /*
 73  * Like mbstowcs(3C), but with UCS-2 wchar_t, and
 74  * one extra arg for the byte order flags.
 75  */
 76 static size_t
 77 ndr__mbstowcs_x(uint16_t *wcs, const char *mbs, size_t nwchars, int flags)
 78 {
 79         size_t obytes, mbslen, wcslen;
 80         int err;

 82         /* NULL or empty input is allowed. */
 83         if (mbs == NULL || *mbs == '\0') {
 84                 if (wcs != NULL && nwchars > 0)
 85                         *wcs = 0;
 86                 return (0);
 87         }

 89         /*
 90          * If wcs == NULL, caller just wants the length.
 91          * Convert into some throw-away space.
 92          */
 93         obytes = nwchars * 2;
 94         if (wcs == NULL) {
 95                 if (obytes > sizeof (junk_wcs))
 96                         return ((size_t)-1);
 97                 wcs = junk_wcs;
 98         }

100         mbslen = strlen(mbs);
101         wcslen = nwchars;
102         err = uconv_u8tou16((const uchar_t *)mbs, &mbslen,
103             wcs, &wcslen, flags);
104         if (err != 0)
105                 return ((size_t)-1);

107         if (wcslen < nwchars)
108                 wcs[wcslen] = 0;

110         return (wcslen);
111 }

113 /*
114  * Like wcstombs(3C), but with UCS-2 wchar_t.
115  */
116 size_t
117 ndr__wcstombs(char *mbs, const uint16_t *wcs, size_t nbytes)
118 {
119         size_t mbslen, wcslen;
120         int err;

122         /* NULL or empty input is allowed. */
123         if (wcs == NULL || *wcs == 0) {
124                 if (mbs != NULL && nbytes > 0)
125                         *mbs = '\0';
126                 return (0);
127         }
```

```
129         /*
130          * If mbs == NULL, caller just wants the length.
131          * Convert into some throw-away space.
132          */
133         if (mbs == NULL) {
134                 if (nbytes > sizeof (junk_mbs))
135                         return ((size_t)-1);
136                 mbs = junk_mbs;
137         }

139         wcslen = ndr__wcslen(wcs);
140         mbslen = nbytes;
141         err = uconv_u16tou8(wcs, &wcslen,
142             (uchar_t *)mbs, &mbslen, UCONV_IN_SYSTEM_ENDIAN);
143         if (err != 0)
144                 return ((size_t)-1);

146         if (mbslen < nbytes)
147                 mbs[mbslen] = '\0';

149         return (mbslen);
150 }

152 /*
153  * Like wcslen(3C), but with UCS-2 wchar_t.
154  */
155 size_t
156 ndr__wcslen(const uint16_t *wc)
157 {
158         size_t len = 0;
159         while (*wc++)
160                 len++;
161         return (len);
162 }
```

```
**********************************************************
    1412 Sun Mar 18 01:12:59 2018
new/usr/src/lib/libmlrpc/common/ndr_wchar.h
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

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

/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 */

#ifndef _NDR_WCHAR_H
#define _NDR_WCHAR_H

/*
 * Some ndr_wchar_t support stuff.
 */

#define NDR_MB_CUR_MAX          3
#define NDR_MB_CHAR_MAX         NDR_MB_CUR_MAX
#define NDR_STRING_MAX          4096

size_t ndr__mbstowcs(uint16_t *, const char *, size_t);
size_t ndr__mbstowcs_le(uint16_t *, const char *, size_t);

size_t ndr__wcslen(const uint16_t *);
size_t ndr__wcstombs(char *, const uint16_t *, size_t);

#endif /* _NDR_WCHAR_H */
```

```
**********************************************************
    4456 Sun Mar 18 01:12:59 2018
new/usr/src/lib/libmlrpc/common/ndrtypes.ndl
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 */

#ifndef _NDRTYPES_NDL_
#define _NDRTYPES_NDL_
#ifndef _NDR_TYPES_NDL_
#define _NDR_TYPES_NDL_

/*
 * Type definitions (and related) used in NDL files and the
 * NDL run-time support libraries.  See also: libmlrpc.h
 */

#define TYPEINFO(TYPE)  ndt__##TYPE

#ifdef NDRGEN

#define ALIGN(X)        [align(X)]
#define OPERATION(X)    [operation(X)]
#define IN              [in]
#define OUT             [out]
#define INOUT           [in out]

#define STRING          [string]
#define SIZE_IS(X)      [size_is(X)]

#define SWITCH(X)       [switch_is(X)]
#define CASE(X)         [case(X)]
#define DEFAULT         [default]

#define INTERFACE(X)    [interface(X)]
#define UUID(X)         [uuid(X)]

#define ARG_IS(X)       [arg_is(X)]

#define REFERENCE       [reference]
#define REF             [reference]
```



```
#define UNIQUE          [unique]
#define PTR             [ptr]
#define POINTER_DEFAULT(X)      [pointer_default(X)]

#define ANY_SIZE_ARRAY  *

#define IMPORT_EXTERN   [extern]

#define BYTE    uchar
#define WORD    ushort
#define DWORD   ulong
#define ntstatus_t      ulong

#define LPTSTR  STRING wchar *

#define LPBYTE  uchar *
#define LPWORD  ushort *
#define LPDWORD ulong *

/*
 * Opaque context handle.
 */
#ifndef CONTEXT_HANDLE
#define CONTEXT_HANDLE(NAME)    \
        struct NAME {           \
                DWORD data1;    \
                DWORD data2;    \
                WORD  data3[2]; \
                BYTE  data4[8]; \
        };                      \
        typedef struct NAME
#endif /* CONTEXT_HANDLE */

#define EXTERNTYPEINFO(TYPE)

#else /* NDRGEN */

#define ALIGN(X)
#define OPERATION(X)
#define IN
#define OUT
#define INOUT

#define STRING
#define SIZE_IS(X)

#define SWITCH(X)
#define CASE(X)
#define DEFAULT

#define INTERFACE(X)
#define UUID(X)

#define ARG_IS(X)

#define REFERENCE
#define REF
#define UNIQUE
#define PTR
#define POINTER_DEFAULT(X)

#define IMPORT_EXTERN

/*
 * When not using ndrgen, get BYTE, WORD, DWORD definitions from wintypes.h.
 */
```

```
112 #include <smb/wintypes.h>
119 #include <smbsrv/wintypes.h>

114 #define EXTERNTYPEINFO(TYPE)    extern struct ndr_typeinfo TYPEINFO(TYPE);


117 /*
118  ************************************************************************
119  * There is a bug in the way that midl and the marshalling code handles
120  * unions so we need to fix some of the data offsets at runtime. The
121  * following macros and the fixup function handle the correction.
122  ************************************************************************
123  */

125 /*
126  * DECL_FIXUP_STRUCT allows us to declare external references to data
127  * structures generated by ndrgen in the _ndr.c file.
128  */
129 #define DECL_FIXUP_STRUCT(NAME) extern struct ndr_typeinfo ndt__##NAME

131 /*
132  * CASE_INFO_ENT is intended to simplify the declaration of the case
133  * statement in the fixup function. Assuming you have followed the
134  * convention for naming the individual structures all you have to do
135  * is add a single line to the fixup function for each new case.
136  */
137 #define CASE_INFO_ENT(NAME,N) \
138                         case N: size1 = sizeof (struct NAME##N); \
139                         break

141 /*
142  * FIXUP_PDU_SIZE is used to patch the appropriate structures (identified
143  * by DECL_FIXUP_STRUCT) at runtime. The values are based on the
144  * switch_index.
145  */
146 #define FIXUP_PDU_SIZE(NAME,SIZE) { \
147         ndt__##NAME.pdu_size_fixed_part = SIZE; \
148         ndt__##NAME.c_size_fixed_part = SIZE; \
149 }


152 #endif /* NDRGEN */

154 /*
155  * UNION_INFO_ENT is intended to simplify adding new entries to a union.
156  * If the entry structures are named using the form FunctionNameX,
157  * where X is the sitch_value, you can just add a single line. Note
158  * that you must also update the fixup function in mlsvc_xxx.c.
159  */
160 #define UNION_INFO_ENT(N,NAME) CASE(N) struct NAME##N info##N
161 #define UNION_INFO_PTR(N,NAME) CASE(N) struct NAME##N *info##N

163 /*
164  * Opaque context handle.
165  */
166 #ifndef CONTEXT_HANDLE
167 #define CONTEXT_HANDLE(NAME)    \
168         struct NAME {           \
169                 DWORD data1;    \
170                 DWORD data2;    \
171                 WORD  data3[2]; \
172                 BYTE  data4[8]; \
173         };                      \
174         typedef struct NAME
175 #endif /* CONTEXT_HANDLE */
```



```
177 #endif /* _NDRTYPES_NDL_ */
170 #endif /* _NDR_TYPES_NDL_ */
```

****
20200 Sun Mar 18 01:12:59 2018
new/usr/src/lib/libmlrpc/common/rpcpdu.ndl
1575 untangle libmlrpc ... (libmlrpc)
****

```
 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) 2007, 2010, Oracle and/or its affiliates. All rights reserved.
23  * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
24  */

26 #ifndef _RPCPDU_NDL_
27 #define _RPCPDU_NDL_

29 #include "ndrtypes.ndl"

31 /*
32  * Normally, constructs are (un)marshalled atoms first, then
33  * constructs, then pointers. This can be confusing sometimes
34  * when debugging. We know that everything in here can be
35  * safely (un)marshalled in member order, so we say so.
36  */
37 #ifdef NDRGEN
38 #define _NO_REORDER_ [_no_reorder]
39 #else
40 #define _NO_REORDER_
41 #endif

43 #define NDR_TRANSFER_SYNTAX_UUID        "8a885d04-1ceb-11c9-9fe8-08002b104860"

45 /*
46  * UUID (Universal Unique IDentifier)
47  */
48 /* (X/Open CAE Spec Appendix A) */
49 struct ndr_dce_uuid {
50         DWORD           time_low;
51         WORD            time_mid;
52         WORD            time_hi_and_version;
53         BYTE            clock_seq_hi_and_reserved;
54         BYTE            clock_seq_low;
55         BYTE            node[6];
56 };
```

_____***unchanged_portion_omitted_***

```
**********************************************************
     976 Sun Mar 18 01:12:59 2018
new/usr/src/lib/libmlrpc/i386/Makefile
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
   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 2007 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #
  25 # ident "%Z%%M% %I%     %E% SMI"
  26 #

  26 include ../Makefile.com

  30 DYNFLAGS +=     -R/usr/lib/smbsrv

  28 install: all $(ROOTLIBS) $(ROOTLINKS) $(ROOTLINT)
```

```
**********************************************************
     976 Sun Mar 18 01:12:59 2018
new/usr/src/lib/libmlrpc/sparc/Makefile
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
   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 2007 Sun Microsystems, Inc.  All rights reserved.
  23 # Use is subject to license terms.
  24 #
  25 # ident "%Z%%M% %I%     %E% SMI"
  26 #

  26 include ../Makefile.com

  30 DYNFLAGS +=     -R/usr/lib/smbsrv

  28 install: all $(ROOTLIBS) $(ROOTLINKS) $(ROOTLINT)
```

```
**********************************************************
    1063 Sun Mar 18 01:13:00 2018
new/usr/src/lib/libmlrpc/sparcv9/Makefile
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# ident "%Z%%M% %I%     %E% SMI"
#

MACH_LDLIBS +=  -L$(ROOT)/usr/lib/smbsrv/$(MACH64)

include ../Makefile.com
include ../../Makefile.lib.64
include ../../../Makefile.lib.64

DYNFLAGS +=     -R/usr/lib/smbsrv/$(MACH64)

install: all $(ROOTLIBS64) $(ROOTLINKS64) $(ROOTLINT64)
```

```
**********************************************************
    6950 Sun Mar 18 01:13:00 2018
new/usr/src/lib/smbsrv/libmlsvc/common/mlsvc_client.c
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

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

/*
 * Copyright (c) 2007, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright 2015 Nexenta Systems, Inc.  All rights reserved.
 */

/*
 * Client NDR RPC interface.
 */

#include <sys/types.h>
#include <sys/errno.h>
#include <sys/fcntl.h>
#include <time.h>
#include <strings.h>
#include <assert.h>
#include <errno.h>
#include <thread.h>
#include <syslog.h>
#include <synch.h>

#include <netsmb/smbfs_api.h>
#include <smbsrv/libsmb.h>
#include <smbsrv/libsmbns.h>
#include <smbsrv/libmlrpc.h>
#include <smbsrv/libmlsvc.h>
#include <smbsrv/ndl/srvsvc.ndl>
#include <libsmbrdr.h>
#include <mlsvc.h>

static int ndr_xa_init(ndr_client_t *, ndr_xa_t *);
static int ndr_xa_exchange(ndr_client_t *, ndr_xa_t *);
static int ndr_xa_read(ndr_client_t *, ndr_xa_t *);
static void ndr_xa_preserve(ndr_client_t *, ndr_xa_t *);
static void ndr_xa_destruct(ndr_client_t *, ndr_xa_t *);
static void ndr_xa_release(ndr_client_t *);


/*
 * This call must be made to initialize an RPC client structure and bind
 * to the remote service before any RPCs can be exchanged with that service.
```



```
 *
 * The mlsvc_handle_t is a wrapper that is used to associate an RPC handle
 * with the client context for an instance of the interface.  The handle
 * is zeroed to ensure that it doesn't look like a valid handle -
 * handle content is provided by the remove service.
 *
 * The client points to this top-level handle so that we know when to
 * unbind and teardown the connection.  As each handle is initialized it
 * will inherit a reference to the client context.
 *
 * Returns 0 or an NT_STATUS:
 *      NT_STATUS_BAD_NETWORK_PATH      (get server addr)
 *      NT_STATUS_NETWORK_ACCESS_DENIED (connect, auth)
 *      NT_STATUS_BAD_NETWORK_NAME      (tcon, open)
 *      NT_STATUS_ACCESS_DENIED         (open pipe)
 *      NT_STATUS_INVALID_PARAMETER     (rpc bind)
 *
 *      NT_STATUS_INTERNAL_ERROR        (bad args etc)
 *      NT_STATUS_NO_MEMORY
 */
DWORD
ndr_rpc_bind(mlsvc_handle_t *handle, char *server, char *domain,
    char *username, const char *service)
{
        struct smb_ctx          *ctx = NULL;
        ndr_client_t            *clnt = NULL;
        ndr_service_t           *svc;
        DWORD                   status;
        int                     fd = -1;
        int                     rc;

        if (handle == NULL || server == NULL || server[0] == '\0' ||
            domain == NULL || username == NULL)
                return (NT_STATUS_INTERNAL_ERROR);

        /* In case the service was not registered... */
        if ((svc = ndr_svc_lookup_name(service)) == NULL)
                return (NT_STATUS_INTERNAL_ERROR);

        /*
         * Some callers pass this when they want a NULL session.
         * Todo: have callers pass an empty string for that.
         */
        if (strcmp(username, MLSVC_ANON_USER) == 0)
                username = "";

        /*
         * Setup smbfs library handle, authenticate, connect to
         * the IPC$ share.  This will reuse an existing connection
         * if the driver already has one for this combination of
         * server, user, domain.  It may return any of:
         *      NT_STATUS_BAD_NETWORK_PATH      (get server addr)
         *      NT_STATUS_NETWORK_ACCESS_DENIED (connect, auth)
         *      NT_STATUS_BAD_NETWORK_NAME      (tcon)
         */
        status = smbrdr_ctx_new(&ctx, server, domain, username);
        if (status != NT_STATUS_SUCCESS) {
                syslog(LOG_ERR, "ndr_rpc_bind: smbrdr_ctx_new"
                    "(Srv=%s Dom=%s User=%s), %s (0x%x)",
                    server, domain, username,
                    xlate_nt_status(status), status);
                /* Tell the DC Locator this DC failed. */
                smb_ddiscover_bad_dc(server);
                goto errout;
        }
```

```
121         /*
122          * Open the named pipe.
123          */
124         fd = smb_fh_open(ctx, svc->endpoint, O_RDWR);
125         if (fd < 0) {
126                 rc = errno;
127                 syslog(LOG_DEBUG, "ndr_rpc_bind: "
128                     "smb_fh_open (%s) err=%d",
129                     svc->endpoint, rc);
130                 switch (rc) {
131                 case EACCES:
132                         status = NT_STATUS_ACCESS_DENIED;
133                         break;
134                 default:
135                         status = NT_STATUS_BAD_NETWORK_NAME;
136                         break;
137                 }
138                 goto errout;
139         }

141         /*
142          * Setup the RPC client handle.
143          */
144         if ((clnt = malloc(sizeof (ndr_client_t))) == NULL) {
145                 status = NT_STATUS_NO_MEMORY;
146                 goto errout;
147         }
148         bzero(clnt, sizeof (ndr_client_t));

150         clnt->handle = &handle->handle;
151         clnt->xa_init = ndr_xa_init;
152         clnt->xa_exchange = ndr_xa_exchange;
153         clnt->xa_read = ndr_xa_read;
154         clnt->xa_preserve = ndr_xa_preserve;
155         clnt->xa_destruct = ndr_xa_destruct;
156         clnt->xa_release = ndr_xa_release;
157         clnt->xa_private = ctx;
158         clnt->xa_fd = fd;

160         ndr_svc_binding_pool_init(&clnt->binding_list,
161             clnt->binding_pool, NDR_N_BINDING_POOL);

163         if ((clnt->heap = ndr_heap_create()) == NULL) {
164                 status = NT_STATUS_NO_MEMORY;
165                 goto errout;
166         }

168         /*
169          * Fill in the caller's handle.
170          */
171         bzero(&handle->handle, sizeof (ndr_hdid_t));
172         handle->clnt = clnt;

174         /*
175          * Do the OtW RPC bind.
176          */
177         rc = ndr_clnt_bind(clnt, service, &clnt->binding);
178         switch (rc) {
179         case NDR_DRC_FAULT_OUT_OF_MEMORY:
180                 status = NT_STATUS_NO_MEMORY;
181                 break;
182         case NDR_DRC_FAULT_API_SERVICE_INVALID: /* not registered */
183                 status = NT_STATUS_INTERNAL_ERROR;
184                 break;
185         default:
186                 if (NDR_DRC_IS_FAULT(rc)) {
```



```
187                         status = NT_STATUS_INVALID_PARAMETER;
188                         break;
189                 }
190                 /* FALLTHROUGH */
191         case NDR_DRC_OK:
192                 return (NT_STATUS_SUCCESS);
193         }

195         syslog(LOG_DEBUG, "ndr_rpc_bind: "
196             "ndr_clnt_bind, %s (0x%x)",
197             xlate_nt_status(status), status);

199 errout:
200         handle->clnt = NULL;
201         if (clnt != NULL) {
202                 ndr_heap_destroy(clnt->heap);
203                 free(clnt);
204         }
205         if (ctx != NULL) {
206                 if (fd != -1)
207                         (void) smb_fh_close(fd);
208                 smbrdr_ctx_free(ctx);
209         }

211         return (status);
212 }
```

_____unchanged_portion_omitted_

```
247 /*
248  * Call the RPC function identified by opnum.  The remote service is
249  * identified by the handle, which should have been initialized by
250  * ndr_rpc_bind.
251  *
252  * If the RPC call is successful (returns 0), the caller must call
253  * ndr_rpc_release to release the heap.  Otherwise, we release the
254  * heap here.
255  */
256 int
257 ndr_rpc_call(mlsvc_handle_t *handle, int opnum, void *params)
258 {
259         ndr_client_t *clnt = handle->clnt;
260         int rc;

262         if (ndr_rpc_get_heap(handle) == NULL)
263                 return (-1);

265         rc = ndr_clnt_call(clnt->binding, opnum, params);

267         /*
268          * Always clear the nonull flag to ensure
269          * it is not applied to subsequent calls.
270          */
271         clnt->nonull = B_FALSE;

273         if (NDR_DRC_IS_FAULT(rc)) {
274                 ndr_rpc_release(handle);
275                 return (-1);
276         }

278         return (0);
279 }

281 /*
282  * Outgoing strings should not be null terminated.
283  */
240 void
```

```
285 ndr_rpc_set_nonull(mlsvc_handle_t *handle)
286 {
287         handle->clnt->nonull = B_TRUE;
288 }

290 /*
291  * Get the session key from a bound RPC client handle.
292  *
293  * The key returned is the 16-byte "user session key"
294  * established by the underlying authentication protocol
295  * (either Kerberos or NTLM).  This key is needed for
296  * SAM RPC calls such as SamrSetInformationUser, etc.
297  * See [MS-SAMR] sections: 2.2.3.3, 2.2.7.21, 2.2.7.25.
298  *
299  * Returns zero (success) or an errno.
300  */
301 int
302 ndr_rpc_get_ssnkey(mlsvc_handle_t *handle,
303         unsigned char *ssn_key, size_t len)
304 {
305         ndr_client_t *clnt = handle->clnt;
306         int rc;

308         if (clnt == NULL)
309                 return (EINVAL);

311         rc = smb_fh_getssnkey(clnt->xa_fd, ssn_key, len);
312         return (rc);
313 }

315 void *
316 ndr_rpc_malloc(mlsvc_handle_t *handle, size_t size)
317 {
318         ndr_heap_t *heap;

320         if ((heap = ndr_rpc_get_heap(handle)) == NULL)
321                 return (NULL);

323         return (ndr_heap_malloc(heap, size));
324 }

326 ndr_heap_t *
327 ndr_rpc_get_heap(mlsvc_handle_t *handle)
328 {
329         ndr_client_t *clnt = handle->clnt;

331         if (clnt->heap == NULL)
332                 clnt->heap = ndr_heap_create();

334         return (clnt->heap);
335 }

337 /*
338  * Must be called by RPC clients to free the heap after a successful RPC
339  * call, i.e. ndr_rpc_call returned 0.  The caller should take a copy
340  * of any data returned by the RPC prior to calling this function because
341  * returned data is in the heap.
342  */
343 void
344 ndr_rpc_release(mlsvc_handle_t *handle)
345 {
346         ndr_client_t *clnt = handle->clnt;

348         if (clnt->heap_preserved)
349                 ndr_clnt_free_heap(clnt);
350         else
```



```
351                 ndr_heap_destroy(clnt->heap);

353         clnt->heap = NULL;
354 }

356 /*
357  * Returns true if the handle is null.
358  * Otherwise returns false.
359  */
360 boolean_t
361 ndr_is_null_handle(mlsvc_handle_t *handle)
362 {
363         static ndr_hdid_t zero_handle;

365         if (handle == NULL || handle->clnt == NULL)
366                 return (B_TRUE);

368         if (!memcmp(&handle->handle, &zero_handle, sizeof (ndr_hdid_t)))
369                 return (B_TRUE);

371         return (B_FALSE);
372 }

374 /*
375  * Returns true if the handle is the top level bind handle.
376  * Otherwise returns false.
377  */
378 boolean_t
379 ndr_is_bind_handle(mlsvc_handle_t *handle)
380 {
381         return (handle->clnt->handle == &handle->handle);
382 }

384 /*
385  * Pass the client reference from parent to child.
386  */
387 void
388 ndr_inherit_handle(mlsvc_handle_t *child, mlsvc_handle_t *parent)
389 {
390         child->clnt = parent->clnt;
391 }

393 void
241 ndr_rpc_status(mlsvc_handle_t *handle, int opnum, DWORD status)
242 {
243         ndr_service_t *svc;
244         char *name = "NDR RPC";
245         char *s = "unknown";

247         switch (NT_SC_SEVERITY(status)) {
248         case NT_STATUS_SEVERITY_SUCCESS:
249                 s = "success";
250                 break;
251         case NT_STATUS_SEVERITY_INFORMATIONAL:
252                 s = "info";
253                 break;
254         case NT_STATUS_SEVERITY_WARNING:
255                 s = "warning";
256                 break;
257         case NT_STATUS_SEVERITY_ERROR:
258                 s = "error";
259                 break;
260         }

262         if (handle) {
263                 svc = handle->clnt->binding->service;
```

```
264                 name = svc->name;
265         }

267         smb_tracef("%s[0x%02x]: %s: %s (0x%08x)",
268             name, opnum, s, xlate_nt_status(status), status);
422 }

424 /*
425  * The following functions provide the client callback interface.
426  * If the caller hasn't provided a heap, create one here.
427  */
428 static int
429 ndr_xa_init(ndr_client_t *clnt, ndr_xa_t *mxa)
430 {
431         ndr_stream_t    *recv_nds = &mxa->recv_nds;
432         ndr_stream_t    *send_nds = &mxa->send_nds;
433         ndr_heap_t      *heap = clnt->heap;
434         int             rc;

436         if (heap == NULL) {
437                 if ((heap = ndr_heap_create()) == NULL)
438                         return (-1);

440                 clnt->heap = heap;
441         }

443         mxa->heap = heap;

445         rc = nds_initialize(send_nds, 0, NDR_MODE_CALL_SEND, heap);
446         if (rc == 0)
447                 rc = nds_initialize(recv_nds, NDR_PDU_SIZE_HINT_DEFAULT,
448                     NDR_MODE_RETURN_RECV, heap);

450         if (rc != 0) {
451                 nds_destruct(&mxa->recv_nds);
452                 nds_destruct(&mxa->send_nds);
453                 ndr_heap_destroy(mxa->heap);
454                 mxa->heap = NULL;
455                 clnt->heap = NULL;
456                 return (-1);
457         }

459         if (clnt->nonull)
460                 NDS_SETF(send_nds, NDS_F_NONULL);

462         return (0);
463 }

465 /*
466  * This is the entry pointy for an RPC client call exchange with
467  * a server, which will result in an smbrdr SmbTransact request.
468  *
469  * SmbTransact should return the number of bytes received, which
470  * we record as the PDU size, or a negative error code.
471  */
472 static int
473 ndr_xa_exchange(ndr_client_t *clnt, ndr_xa_t *mxa)
474 {
475         ndr_stream_t *recv_nds = &mxa->recv_nds;
476         ndr_stream_t *send_nds = &mxa->send_nds;
477         int err, more, nbytes;

479         nbytes = recv_nds->pdu_max_size;
480         err = smb_fh_xactnp(clnt->xa_fd,
481             send_nds->pdu_size, (char *)send_nds->pdu_base_offset,
482             &nbytes, (char *)recv_nds->pdu_base_offset, &more);
```



```
483         if (err) {
484                 recv_nds->pdu_size = 0;
485                 return (-1);
486         }

488         recv_nds->pdu_size = nbytes;
489         return (0);
490 }

492 /*
493  * This entry point will be invoked if the xa-exchange response contained
494  * only the first fragment of a multi-fragment response.  The RPC client
495  * code will then make repeated xa-read requests to obtain the remaining
496  * fragments, which will result in smbrdr SmbReadX requests.
497  *
498  * SmbReadX should return the number of bytes received, in which case we
499  * expand the PDU size to include the received data, or a negative error
500  * code.
501  */
502 static int
503 ndr_xa_read(ndr_client_t *clnt, ndr_xa_t *mxa)
504 {
505         ndr_stream_t *nds = &mxa->recv_nds;
506         int len;
507         int nbytes;

509         if ((len = (nds->pdu_max_size - nds->pdu_size)) < 0)
510                 return (-1);

512         nbytes = smb_fh_read(clnt->xa_fd, 0, len,
513             (char *)nds->pdu_base_offset + nds->pdu_size);

515         if (nbytes < 0)
516                 return (-1);

518         nds->pdu_size += nbytes;

520         if (nds->pdu_size > nds->pdu_max_size) {
521                 nds->pdu_size = nds->pdu_max_size;
522                 return (-1);
523         }

525         return (nbytes);
526 }

528 /*
529  * Preserve the heap so that the client application has access to data
530  * returned from the server after an RPC call.
531  */
532 static void
533 ndr_xa_preserve(ndr_client_t *clnt, ndr_xa_t *mxa)
534 {
535         assert(clnt->heap == mxa->heap);

537         clnt->heap_preserved = B_TRUE;
538         mxa->heap = NULL;
539 }

541 /*
542  * Dispose of the transaction streams.  If the heap has not been
543  * preserved, we can destroy it here.
544  */
545 static void
546 ndr_xa_destruct(ndr_client_t *clnt, ndr_xa_t *mxa)
547 {
548         nds_destruct(&mxa->recv_nds);
```

```
549         nds_destruct(&mxa->send_nds);

551         if (!clnt->heap_preserved) {
552                 ndr_heap_destroy(mxa->heap);
553                 mxa->heap = NULL;
554                 clnt->heap = NULL;
555         }
556 }

558 /*
559  * Dispose of a preserved heap.
560  */
561 static void
562 ndr_xa_release(ndr_client_t *clnt)
563 {
564         if (clnt->heap_preserved) {
565                 ndr_heap_destroy(clnt->heap);
566                 clnt->heap = NULL;
567                 clnt->heap_preserved = B_FALSE;
568         }
269 }
```

_____unchanged_portion_omitted_

```
**********************************************************
    2675 Sun Mar 18 01:13:00 2018
new/usr/src/tools/ndrgen/ndrgen.sh
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

```
   1 #!/bin/ksh -p
   2 #
   3 # CDDL HEADER START
   4 #
   5 # The contents of this file are subject to the terms of the
   6 # Common Development and Distribution License (the "License").
   7 # You may not use this file except in compliance 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 2009 Sun Microsystems, Inc.  All rights reserved.
  25 # Use is subject to license terms.
  26 #
  27 # Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
  28 #

  30 # This is a wrapper script around the ndrgen compiler (ndrgen1).
  28 # CC must be defined in the environment or on the command line.

  32 NDRPROG="${0}1"
  30 NDRPROG="${0%/*}/ndrgen1"
  31 INCDIR=${ROOT}/usr/include/smbsrv

  34 # Note: most *.ndl files require an ANSI-compatible cpp,
  35 # so we can NOT use /usr/lib/cpp or /usr/ccs/lib/cpp
  36 # Wish there was an easier way to get an ANSI cpp.
  37 CPP="${CC} -E"
  38 CPPFLAGS="-DNDRGEN"
  39 V_FLAG=

  41 PROGNAME=‘basename $0‘

  43 ndrgen_usage()
  44 {
  45         if [[ $1 != "" ]] ; then
  46                 print "$PROGNAME: ERROR: $1"
  47         fi

  49         echo "usage: $PROGNAME [options] file.ndl [file.ndl]..."
  50         echo "  options: -Y cc-path -Ddefine -Iinclude"
  41         echo "usage: $PROGNAME [-Y cpp-path] file [file]..."
  51         exit 1
  52 }

  54 # Process the input ndl file ($1) generating C code on stdout.
  55 process()
  45 # Copy header text from the input ndl file to the generated ndr C file.
  46 ndrgen_copy_header()
```



```
  56 {
  48         ndl_file=$1
  49         ndr_file=$2

  58         # Put the standard top matter
  59         #
  60         # Want the include directive to have just
  61         # include "file.ndl" (no path) so...
  62         inc_ndl=‘basename $1‘
  63         cat - << EOF
  64 /*
  65  * Please do not edit this file.
  66  * It was generated using ndrgen.
  67  */

  69 #include <strings.h>
  70 #include <libmlrpc/ndr.h>
  71 #include "$inc_ndl"
  72 EOF

  74         # Put optional custom top matter
  75         nawk ’BEGIN { copy=0; }
  76         /^\/\* NDRGEN_HEADER_BEGIN \*\// { copy=1; next; }
  77         /^\/\* NDRGEN_HEADER_END \*\// { copy=0; next; }
  78         /./ { if (copy==1) print; }’ $1

  80         # now the real ndrgen output
  81         [ -n "$V_FLAG" ] &&
  82           echo "$CPP $CPPFLAGS $1 | $NDRPROG" >&2
  83         $CPP $CPPFLAGS $1 | $NDRPROG
  54         /./ { if (copy==1) print; }’ < $ndl_file > $ndr_file
  84 }

  57 if [[ $# -lt 1 ]] ; then
  58         ndrgen_usage
  59 fi

  87 while getopts "D:I:Y:V" FLAG
  88 do
  61 while getopts "Y" FLAG $*; do
  89         case $FLAG in
  90         D|I)    CPPFLAGS="$CPPFLAGS -${FLAG}${OPTARG}";;
  91         Y)      CPP="$OPTARG";;
  92         V)      V_FLAG="V";;
  93         *)      ndrgen_usage;;
  63         Y)
  64                 CC_FLAG="y"
  65                 ;;
  66         *)
  67                 ndrgen_usage
  68                 ;;
  94         esac
  95 done
  96 shift $(($OPTIND - 1))

  98 if [[ $# -lt 1 ]] ; then
  99         ndrgen_usage
  72 if [[ $CC_FLAG = "y" ]] ; then
  73         shift $(($OPTIND - 1))

  75         if [[ $# -lt 1 ]] ; then
  76                 ndrgen_usage "C pre-processor path is missing"
  77         else
  78                 CC=$1
  79                 shift $(($OPTIND - 1))
```

```
                # Check for cw being invoked with -_cc or -_gcc
                if [[ $1 = "-_cc" || $1 = "-_gcc" ]] ; then
                        CC_ARG=$1
                        shift $(($OPTIND - 1))
                fi
        fi
fi

if [ ! -x $CPP ] ; then
        ndrgen_usage "cannot run $CPP"
if [[ $CC = "" ]] ; then
        ndrgen_usage "C pre-processor is not defined"
fi

if [ ! -f $CC ] || [ ! -x $CC ] ; then
        ndrgen_usage "cannot run $CC"
fi

for i
do
        if [[ ! -r $i ]] ; then
                print "$PROGNAME: ERROR: cannot read $i"
                exit 1
        fi

        base=`basename $i .ndl`
        process $i > ${base}_ndr.c || {
          echo "ndrgen error";
          rm ${base}_ndr.c;
        }
        BASENAME=`basename $i .ndl`
        TMP_NAME=$BASENAME.ndl.c

        cp $i $TMP_NAME

        if $CC $CC_ARG -E  -D__a64 -D__EXTENSIONS__ -D_FILE_OFFSET_BITS=64 \
                -I. -I${INCDIR} -I${INCDIR}/ndl -DNDRGEN $TMP_NAME | \
                $NDRPROG > $BASENAME.raw
        then
                touch ${BASENAME}_ndr.c
                ndrgen_copy_header $i ${BASENAME}_ndr.c

                cat - << EOF >> ${BASENAME}_ndr.c
/*
 * Please do not edit this file.
 * It was generated using ndrgen.
 */

#include <strings.h>
#include <smbsrv/ndr.h>
#include <smbsrv/ndl/$BASENAME.ndl>
EOF

                cat $BASENAME.raw >> ${BASENAME}_ndr.c

                rm -f $BASENAME.raw
                rm -f $TMP_NAME
        else
                rm -f $BASENAME.raw
                rm -f $TMP_NAME
                exit 1
        fi
done
```

```
**********************************************************
    6639 Sun Mar 18 01:13:00 2018
new/usr/src/tools/quick/make-smbclnt
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

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

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

# Use distributed make (dmake) by default.
make=${MAKE:-dmake}

CLOSED_IS_PRESENT=no
export CLOSED_IS_PRESENT

# Do this if you want to use dbx or gdb
# export SOURCEDEBUG=yes

[ -n "$SRC" ] || {
  echo "SRC not set.  Run 'ws' or 'bldenv' first."
  exit 1
}

cpu='uname -p'
case $cpu in
i386)
        x=intel
        kmdb_arch="amd64"
        mdb_arch="ia32 amd64"
        arch64=amd64
        ;;
sparc)
        x=sparc
        kmdb_arch=v9
        mdb_arch="v7 v9"
        arch64=sparcv9
        ;;
*)  echo "Huh?" ; exit 1;;
esac

#########################################################################

build_tools() {
  test -f $SRC/tools/proto/root_i386-nd/opt/onbld/bin/genoffsets ||
    (cd $SRC/tools && $make install)
  (cd $SRC/common/mapfiles; $make install)
}

clobber_tools() {
  (cd $SRC/tools && $make clobber)
  (cd $SRC/common/mapfiles; $make clobber)
}

#########################################################################
```



```
do_hdrs() {

targ=$1
if [ "$targ" = clobber ]
then
  (cd $SRC/uts && $make -k clobber_h)
  (cd $SRC/head && $make clobber)
fi

if [ "$targ" = install ]
then
  targ=install_h

  # Just the parts of "make sgs" we need, and
  # skip them if they appear to be done.
  # ... stuff under $SRC
  test -f $SRC/uts/common/sys/priv_names.h ||
    (cd $SRC/uts && $make -k all_h)

  test -f $SRC/head/rpcsvc/nispasswd.h ||
    (cd $SRC/head && $make -k install_h)

  # ... stuff under $ROOT (proto area)
  test -d $ROOT/usr/include/sys ||
    (cd $SRC && $make rootdirs)
  test -f $ROOT/usr/include/sys/types.h ||
    (cd $SRC/uts && $make -k install_h)
  test -f $ROOT/usr/include/rpcsvc/daemon_utils.h ||
    (cd $SRC/head && $make install_h)

  # always update the netsmb headers to be safe
  (cd $SRC/uts/common/sys && $make -k install_h)

fi

# Need some library headers too...
for lib in \
  libcryptoutil \
  libpam \
  libsec \
  libshare \
  libsmbfs \
  passwdutil
do
  (cd $SRC/lib/$lib && $make $targ)
done
}

#########################################################################

do_kern() {
  case $1 in
  lint) targ=modlintlib ;;
  *) targ=$1 ;;
  esac
  ( unset SOURCEDEBUG ;
  (cd $SRC/uts/$x/nsmb && $make $targ) ;
  (cd $SRC/uts/$x/smbfs && $make $targ) )
}

#########################################################################

# Note lib1 builds prerequisite libraries not delivered by the
# tar file we create below.  To accelerate clean/install, we
# skip these on clean (but still nuke them for clobber)
```

```
do_lib1() {
  :
}

# lib2 builds stuff we include in the tar file,
# or that we don't mind rebuilding after clean.

do_lib2() {

(cd $SRC/lib/libsmbfs && $make $1)
[ "$1" = install ] &&
  (cd $SRC/lib/libsmbfs && $make _msg)
(cd $SRC/lib/libmlrpc && $make $1)
(cd $SRC/lib/libshare && $make $1 PLUGINS=smbfs)
(cd $SRC/lib/passwdutil && $make $1)
(cd $SRC/lib/pam_modules/smbfs && $make $1)

}

#################################################################

do_cmds() {

case $1 in
install)
  # mount programs need fslib.o
  (cd $SRC/cmd/fs.d && $make fslib.o)
  (cd $SRC/cmd/fs.d/smbclnt && $make $1 catalog)
  ;;
clean|clobber)
  (cd $SRC/cmd/fs.d/smbclnt && $make $1)
  (cd $SRC/cmd/fs.d && $make ${1}_local)
  ;;
esac

# Build the MDB modules, WITH the linktest
(cd $SRC/cmd/mdb/tools && $make $1)

# kmdb_arch is 64-bit only
for a in $kmdb_arch
do
  case $1 in
  install|lint)
    (cd $SRC/cmd/mdb/$x/$a/kmdb &&
        $make kmdb_modlinktest.o )
    ;;
  clean|clobber)
    (cd $SRC/cmd/mdb/$x/$a/kmdb &&
        $make -k $1 )
    ;;
  esac

  (cd $SRC/cmd/mdb/$x/$a/nsmb &&
        $make $1 KMDB_LINKTEST_ENABLE= )
  (cd $SRC/cmd/mdb/$x/$a/smbfs &&
        $make $1 KMDB_LINKTEST_ENABLE= )
done
}


#################################################################
# This builds $SRC/TAGS (and cscope.files) in a helpful order.

do_tags() {
        (cd $SRC ;
```



```
        find uts/common/sys -name '*.[ch]' -print |sort
        find uts/common/net -name '*.[ch]' -print |sort
        find uts/common/netinet -name '*.[ch]' -print |sort
        find uts/common/smb -name '*.[ch]' -print |sort
        find uts/common/netsmb -name '*.[ch]' -print |sort
        find uts/common/fs/smbclnt -name '*.[ch]' -print |sort
        find head -name '*.h' -print |sort
        find lib/libsmbfs -name '*.[ch]' -print |sort
        find cmd/fs.d/smbclnt -name '*.[ch]' -print |sort
        find common/smbclnt -name '*.[ch]' -print |sort
        ) > $SRC/cscope.files

        (cd $SRC ;
        exctags -e --langmap=c:+.ndl -h ndl -L - < cscope.files
        cscope -b )
}

#################################################################
# This creates a tarfile one can use to update a test machine.

do_tar() {
        git_rev=`git rev-parse --short=8 HEAD`
        files="
lib/svc/manifest/network/smb/client.xml
lib/svc/method/smb-client
opt/smbcl-tests/tests/srvenum
opt/smbcl-tests/tests/srvinfo
opt/smbcl-tests/tests/tconn
usr/bin/smbutil
usr/kernel/drv/$arch64/nsmb
usr/kernel/fs/$arch64/smbfs
usr/kernel/kmdb/$arch64/nsmb
usr/kernel/kmdb/$arch64/smbfs
usr/lib/$arch64/libsmbfs.so.1
usr/lib/fs/smbfs/$arch64/libshare_smbfs.so.1
usr/lib/fs/smbfs/chacl
usr/lib/fs/smbfs/dfshares
usr/lib/fs/smbfs/libshare_smbfs.so.1
usr/lib/fs/smbfs/lsacl
usr/lib/fs/smbfs/mount
usr/lib/fs/smbfs/share
usr/lib/fs/smbfs/umount
usr/lib/fs/smbfs/unshare
usr/lib/libmlrpc.so.1
usr/lib/libsmbfs.so.1
usr/lib/mdb/kvm/$arch64/nsmb.so
usr/lib/mdb/kvm/$arch64/smbfs.so
usr/lib/mdb/kvm/nsmb.so
usr/lib/mdb/kvm/smbfs.so
usr/lib/security/$arch64/pam_smbfs_login.so.1
usr/lib/security/pam_smbfs_login.so.1
usr/lib/smbfs/smbiod
usr/lib/smbfs/smbiod-svc
"

        (cd $ROOT && tar cfj ../../smbclnt-${git_rev}.tar.bz2 $files)
}

#################################################################

if [ "$1" = "" ]; then
  set '?' # force usage
fi

set -x
```

```
for arg
do
  case "$arg" in
  install)
    build_tools
    set -e
    do_hdrs $arg
    do_kern $arg
    do_lib1 $arg
    do_lib2 $arg
    do_cmds $arg
    ;;
  lint)
    do_kern $arg
    do_lib1 $arg
    do_lib2 $arg
    do_cmds $arg
    ;;
  clean)
    # intentionally skip: lib1, hdrs, tools
    do_cmds $arg
    do_lib2 $arg
    do_kern $arg
    ;;
  clobber)
    do_cmds $arg
    do_lib2 $arg
    do_lib1 $arg
    do_kern $arg
    do_hdrs $arg
    clobber_tools
    ;;
  tags)
    do_tags
    ;;
  tar)
    do_tar
    ;;
  *)
    echo "Usage: $0 {install|lint|clean|clobber|tags|tar}";
    exit 1;
    ;;
  esac
done
```

```
**********************************************************
    1299 Sun Mar 18 01:13:00 2018
new/usr/src/uts/common/smb/Makefile
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #

  22 #
  23 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
  24 # Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
  25 #

  27 include ../../../Makefile.master

  29 HDRS=   \
  30         doserror.h              \
  31         lmerr.h                 \
  32         nterror.h               \
  33         ntstatus.h              \
  34         wintypes.h
  32         ntstatus.h


  37 ROOTDIR= $(ROOT)/usr/include/smb
  38 ROOTHDRS= $(HDRS:%=$(ROOTDIR)/%)
  39 CHECKHDRS= $(HDRS:%.h=%.check)

  41 $(ROOTDIR)/%: %
  42         $(INS.file)

  44 $(ROOTDIR):
  45         $(INS.dir)

  47 .KEEP_STATE:

  49 .PARALLEL: $(CHECKHDRS)

  51 install_h: $(ROOTDIR) $(ROOTHDRS)

  53 check: $(CHECKHDRS)
```

```
**********************************************************
    1756 Sun Mar 18 01:13:00 2018
new/usr/src/uts/common/smb/wintypes.h
1575 untangle libmlrpc ... (libmlrpc)
**********************************************************
```

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

/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 */

#ifndef _SMB_WINTYPES_H
#define _SMB_WINTYPES_H
#ifndef _SMBSRV_WINTYPES_H
#define _SMBSRV_WINTYPES_H

#include <sys/types.h>

/*
 * Standard win32 types and definitions.
 */

#ifdef __cplusplus
extern "C" {
#endif

#ifndef UNSIGNED_TYPES_DEFINED
#define UNSIGNED_TYPES_DEFINED

typedef uint8_t BYTE;
typedef uint16_t WORD;
typedef uint32_t DWORD;
typedef DWORD ntstatus_t;

/* pointers to those types */
typedef BYTE *LPBYTE;
typedef WORD *LPWORD;
typedef DWORD *LPDWORD;

/* Note: Internally, this is always a UTF-8 string. */
typedef uint32_t ntstatus_t;
typedef uint8_t *LPTSTR;
typedef uint8_t *LPBYTE;
typedef uint16_t *LPWORD;
```



```
typedef uint32_t *LPDWORD;

#endif /* UNSIGNED_TYPES_DEFINED */


/* XXX This does not really belong here... */
#ifndef ANY_SIZE_ARRAY
#define ANY_SIZE_ARRAY  1
#endif /* ANY_SIZE_ARRAY */

/* CONTEXT_HANDLE now in ndrtypes.ndl */
/*
 * Opaque context handle.
 */
#ifndef CONTEXT_HANDLE
#define CONTEXT_HANDLE(NAME)    \
        struct NAME {           \
                DWORD data1;    \
                DWORD data2;    \
                WORD  data3[2]; \
                BYTE  data4[8]; \
        };                      \
        typedef struct NAME
#endif /* CONTEXT_HANDLE */


#ifdef __cplusplus
}
#endif

#endif /* _SMB_WINTYPES_H */
#endif /* _SMBSRV_WINTYPES_H */
```