****

21739 Mon Apr 15 12:20:21 2019
new/usr/src/uts/Makefile.uts
10805 Fix for 10687 can be improved

****

```
#
# 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) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2011 Bayard G. Bell. All rights reserved.
# Copyright (c) 2011 by Delphix. All rights reserved.
# Copyright (c) 2013 Andrew Stormont.  All rights reserved.
# Copyright 2016 Hans Rosenfeld <rosenfeld@grumpf.hope-2000.org>
# Copyright (c) 2019, Joyent, Inc.
#

#
#       This Makefile contains the common targets and definitions for
#       all kernels. It is to be included in the Makefiles for specific
#       implementation architectures and processor architecture dependent
#       modules: i.e.: all driving kernel Makefiles.
#
#       Include global definitions:
#
include $(SRC)/Makefile.master

#
#       No text domain in the kernel.
#
DTEXTDOM =

#
#       Keep references to $(SRC)/common relative.
COMMONBASE=     $(UTSBASE)/../common

#
#       Setup build-specific vars
#       To add a build type:
#               add name to ALL_BUILDS32 & ALL_BUILDS64
#               set CLASS_name and OBJ_DIR_name
#               add targets to Makefile.targ
#

#
#       DEF_BUILDS is for def, lint, sischeck, and install
#       ALL_BUILDS is for everything else (all, clean, ...)
#
```

```
# The NOT_RELEASE_BUILD noise is to maintain compatibility with the
# gatekeeper’s nightly build script.
#
DEF_BUILDS32                            = obj32
DEF_BUILDS64                            = obj64
DEF_BUILDSONLY64                        = obj64
$(NOT_RELEASE_BUILD)DEF_BUILDS32        = debug32
$(NOT_RELEASE_BUILD)DEF_BUILDS64        = debug64
$(NOT_RELEASE_BUILD)DEF_BUILDSONLY64    = debug64
ALL_BUILDS32                            = obj32 debug32
ALL_BUILDS64                            = obj64 debug64
ALL_BUILDSONLY64                        = obj64 debug64

#
#       For modules in 64b dirs that aren’t built 64b
#       or modules in 64b dirs that aren’t built 32b we
#       need to create empty modlintlib files so global lint works
#
LINT32_BUILDS   = debug32
LINT64_BUILDS   = debug64

#
#       Build class (32b or 64b)
#
CLASS_OBJ32     = 32
CLASS_DBG32     = 32
CLASS_OBJ64     = 64
CLASS_DBG64     = 64
CLASS           = $(CLASS_$(BUILD_TYPE))

#
#       Build subdirectory
#
OBJS_DIR_OBJ32  = obj32
OBJS_DIR_DBG32  = debug32
OBJS_DIR_OBJ64  = obj64
OBJS_DIR_DBG64  = debug64
OBJS_DIR        = $(OBJS_DIR_$(BUILD_TYPE))

#
#       Create defaults so empty rules don’t
#       confuse make
#
CLASS_          = 64
OBJS_DIR_       = debug64

#
#       Build tools
#
CC_sparc_32     = $(sparc_CC)
CC_sparc_64     = $(sparcv9_CC)

CC_i386_32      = $(i386_CC)
CC_i386_64      = $(amd64_CC)
CC_amd64_64     = $(amd64_CC)

CC              = $(CC_$(MACH)_$(CLASS))

AS_sparc_32     = $(sparc_AS)
AS_sparc_64     = $(sparcv9_AS)

AS_i386_32      = $(i386_AS)
AS_i386_64      = $(amd64_AS)
AS_amd64_64     = $(amd64_AS)

AS              = $(AS_$(MACH)_$(CLASS))
```

```
LD_sparc_32     = $(sparc_LD)
LD_sparc_64     = $(sparcv9_LD)

LD_i386_32      = $(i386_LD)
LD_i386_64      = $(amd64_LD)
LD_amd64_64     = $(amd64_LD)

LD              = $(LD_$(MACH)_$(CLASS))

LINT_sparc_32   = $(sparc_LINT)
LINT_sparc_64   = $(sparcv9_LINT)

LINT_i386_32    = $(i386_LINT)
LINT_i386_64    = $(amd64_LINT)
LINT_amd64_64   = $(amd64_LINT)

LINT            = $(LINT_$(MACH)_$(CLASS))

MODEL_32        = ilp32
MODEL_64        = lp64
MODEL           = $(MODEL_$(CLASS))

#
#       Build rules for linting the kernel.
#
LHEAD = $(ECHO) "\n$@";

# Note: egrep returns "failure" if there are no matches, which is
# exactly the opposite of what we need.
LGREP.2 =       if egrep -v ' (_init|_fini|_info) '; then false; else true; fi

LTAIL =

LINT.c =        $(LINT) -c -dirout=$(LINTS_DIR) $(LINTFLAGS) $(LINT_DEFS) $(CPPF

# Please do not add new erroff directives here.  If you need to disable
# lint warnings in your module for things that cannot be fixed in any
# reasonable manner, please augment LINTTAGS in your module Makefile
# instead.
LINTTAGS        = -erroff=E_INCONS_ARG_DECL2
LINTTAGS        += -erroff=E_INCONS_VAL_TYPE_DECL2

LINTFLAGS_sparc_32      = $(LINTCCMODE) -nsxmuF -errtags=yes
LINTFLAGS_sparc_64      = $(LINTFLAGS_sparc_32) -m64
LINTFLAGS_i386_32       = $(LINTCCMODE) -nsxmuF -errtags=yes
LINTFLAGS_i386_64       = $(LINTFLAGS_i386_32) -m64

LINTFLAGS       = $(LINTFLAGS_$(MACH)_$(CLASS)) $(LINTTAGS)
LINTFLAGS       += $(C99LMODE)

#
#       Override this variable to modify the name of the lint target.
#
LINT_MODULE=    $(MODULE)

#
#       Build the compile/assemble lines:
#
EXTRA_OPTIONS           =
AS_DEFS                 = -D_ASM -D__STDC__=0

ALWAYS_DEFS_32          = -D_KERNEL -D_SYSCALL32 -D_DDI_STRICT
ALWAYS_DEFS_64          = -D_KERNEL -D_SYSCALL32 -D_SYSCALL32_IMPL -D_ELF64 \
                        -D_DDI_STRICT
#
```

```
# XX64  This should be defined by the compiler!
#
ALWAYS_DEFS_64          += -Dsun -D__sun -D__SVR4
ALWAYS_DEFS             = $(ALWAYS_DEFS_$(CLASS))

#
#       CPPFLAGS is deliberatly set with a "=" and not a "+=".  For the kernel
#       the header include path should not look for header files outside of
#       the kernel code.  This "=" removes the search path built in
#       Makefile.master inside CPPFLAGS.  Ditto for AS_CPPFLAGS.
#
CPPFLAGS        = $(ALWAYS_DEFS) $(ALL_DEFS) $(CONFIG_DEFS) \
                   $(INCLUDE_PATH) $(EXTRA_OPTIONS)
ASFLAGS         += -P
AS_CPPFLAGS     = $(ALWAYS_DEFS) $(ALL_DEFS) $(CONFIG_DEFS) $(AS_DEFS) \
                   $(AS_INC_PATH) $(EXTRA_OPTIONS)

#
#       Make it (relatively) easy to share compilation options between
#       all kernel implementations.
#

# Override the default, the kernel is squeaky clean
CERRWARN = -errtags=yes -errwarn=%all

CERRWARN += -_gcc=-Wno-missing-braces
CERRWARN += -_gcc=-Wno-sign-compare
CERRWARN += -_gcc=-Wno-unknown-pragmas
CERRWARN += -_gcc=-Wno-unused-parameter
CERRWARN += -_gcc=-Wno-missing-field-initializers

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

CERRWARN += -_smatch=-p=illumos_kernel
include $(SRC)/Makefile.smatch

#
# Unfortunately, _IOWR() is regularly used with a third argument of 0,
# so we have to disable all these smatch checks.
#
SMOFF += sizeof

#
# DEBUG-only macros that define away to nothing confuse this check,
# unfortunately.
#
$(RELEASE_BUILD)SMOFF += indenting

CSTD = $(CSTD_GNU99)

CFLAGS_uts              =
CFLAGS_uts              += $(STAND_FLAGS_$(CLASS))
CFLAGS_uts              += $(CCVERBOSE)
CFLAGS_uts              += $(ILDOFF)
CFLAGS_uts              += $(XAOPT)
CFLAGS_uts              += $(CTF_FLAGS_$(CLASS))
CFLAGS_uts              += $(CERRWARN)
CFLAGS_uts              += $(CCNOAUTOINLINE)
CFLAGS_uts              += $(CCNOREORDER)
CFLAGS_uts              += $(CCNOAGGRESSIVELOOPS)
CFLAGS_uts              += $(CGLOBALSTATIC)
CFLAGS_uts              += $(EXTRA_CFLAGS)
CFLAGS_uts              += $(CSOURCEDEBUGFLAGS)
CFLAGS_uts              += $(CUSERFLAGS)

#
#       Declare that $(OBJECTS) and $(LINTS) can be compiled in parallel.
#       The DUMMY target is for those instances where OBJECTS and LINTS
#       are empty (to avoid an unconditional .PARALLEL).
.PARALLEL:      $(OBJECTS) $(LINTS) DUMMY

#
#       Expanded dependencies
#
DEF_DEPS        = $(DEF_BUILDS:%=def.%)
ALL_DEPS        = $(ALL_BUILDS:%=all.%)
CLEAN_DEPS      = $(ALL_BUILDS:%=clean.%)
CLOBBER_DEPS    = $(ALL_BUILDS:%=clobber.%)
LINT_DEPS       = $(DEF_BUILDS:%=lint.%)
MODLINTLIB_DEPS = $(DEF_BUILDS:%=modlintlib.%)
MODLIST_DEPS    = $(DEF_BUILDS:%=modlist.%)
CLEAN_LINT_DEPS = $(ALL_BUILDS:%=clean.lint.%)
INSTALL_DEPS    = $(DEF_BUILDS:%=install.%)
SYM_DEPS        = $(SYM_BUILDS:%=symcheck.%)
SISCHECK_DEPS   = $(DEF_BUILDS:%=sischeck.%)
SISCLEAN_DEPS   = $(ALL_BUILDS:%=sisclean.%)

#
#       Default module name
#
BINARY          = $(OBJS_DIR)/$(MODULE)

#
#       Default cleanup definitions
#
CLEANLINTFILES  = $(LINTS) $(MOD_LINT_LIB)
CLEANFILES      = $(OBJECTS) $(CLEANLINTFILES)
CLOBBERFILES    = $(BINARY) $(CLEANFILES)

#
#       Installation constants:
#
#               FILEMODE is the mode given to the kernel modules
#               CFILEMODE is the mode given to the ’.conf’ files
#
FILEMODE         = 755
DIRMODE          = 755
CFILEMODE        = 644

#
#       Special Installation Macros for the installation of ’.conf’ files.
#
#       These are unique because they are not installed from the current
#       working directory.
#
# Sigh.  Apparently at some time in the past there was a confusion on
# whether the name is SRC_CONFFILE or SRC_CONFILE.  Consistency with the
# other names would indicate SRC_CONFFILE, but the voting is >180 Makefiles
# with SRC_CONFILE and about 11 with SRC_CONFFILE.  Software development
# isn’t a popularity contest, though, and so my inclination is to define
# both names for now and incrementally convert to SRC_CONFFILE to be consistent
# with the other names.
#
CONFFILE                = $(MODULE).conf
SRC_CONFFILE            = $(CONF_SRCDIR)/$(CONFFILE)
SRC_CONFILE             = $(SRC_CONFFILE)
ROOT_CONFFILE_32        = $(ROOTMODULE).conf
ROOT_CONFFILE_64        = $(ROOTMODULE:%/$(SUBDIR64)/$(MODULE)=%/$(MODULE)).conf
ROOT_CONFFILE           = $(ROOT_CONFFILE_$(CLASS))
```

```

INS.conffile= \
        $(RM) $@; $(INS) -s -m $(CFILEMODE) -f $(@D) $(SRC_CONFFILE)

#
# The CTF merge of child kernel modules is performed against one of the genunix
# modules.  For Intel builds, all modules will be used with a single genunix:
# the one built in intel/genunix.  For SPARC builds, a given
# module may be
# used with one of a number of genunix files, depending on what platform the
# module is deployed on.  We merge against the sun4u genunix to optimize for
# the common case.  We also merge against the ip driver since networking is
# typically loaded and types defined therein are shared between many modules.
#
CTFMERGE_GUDIR_sparc    = sun4u
CTFMERGE_GUDIR_i386     = intel
CTFMERGE_GUDIR          = $(CTFMERGE_GUDIR_$(MACH))

CTFMERGE_GENUNIX        = \
        $(UTSBASE)/$(CTFMERGE_GUDIR)/genunix/$(OBJS_DIR)/genunix

#
# Used to uniquify a non-genunix module against genunix. $VERSION is used
# for the label.
#
# For the ease of developers dropping modules onto possibly unrelated systems,
# you can set NO_GENUNIX_UNIQUIFY= in the environment to skip uniquifying
# against genunix.
#
NO_GENUNIX_UNIQUIFY=$(POUND_SIGN)
CTFMERGE_GENUNIX_DFLAG=-d $(CTFMERGE_GENUNIX)
$(NO_GENUNIX_UNIQUIFY)CTF_GENUNIX_DFLAG=

CTFMERGE_UNIQUIFY_AGAINST_GENUNIX       = \
        $(CTFMERGE) $(CTFMRGFLAGS) -L VERSION \
        $(CTFMERGE_GENUNIX_DFLAG) -o $@ $(OBJECTS) $(CTFEXTRAOBJS)

#
# Used to merge the genunix module.
#
CTFMERGE_GENUNIX_MERGE          = \
        $(CTFMERGE) $(CTFMRGFLAGS) -L VERSION -o $@ \
        $(OBJECTS) $(CTFEXTRAOBJS) $(IPCTF_TARGET)

#
# We ctfmerge the ip objects into genunix to maximize the number of common types
# found there, thus maximizing the effectiveness of uniquification.  We don’t
# want the genunix build to have to know about the individual ip objects, so we
# put them in an archive.  The genunix ctfmerge then includes this archive.
#
IPCTF           = $(IPDRV_DIR)/$(OBJS_DIR)/ipctf.a

#
# Rule for building fake shared libraries used for symbol resolution
# when building other modules.  -znoreloc is needed here to avoid
# tripping over code that isn’t really suitable for shared libraries.
#
BUILD.SO                = \
        $(LD) -o $@ $(GSHARED) $(ZNORELOC) -h $(SONAME)

#
# SONAME defaults for common fake shared libraries.
#
$(LIBGEN)               := SONAME = $(MODULE)
$(PLATLIB)              := SONAME = misc/platmod
```

```
$(CPULIB)               := SONAME = 'cpu/$$CPU'
$(DTRACESTUBS)          := SONAME = dtracestubs

#
#       Installation directories
#

#
#       For now, 64b modules install into a subdirectory
#       of their 32b brethren.
#
SUBDIR64_sparc          = sparcv9
SUBDIR64_i386           = amd64
SUBDIR64                = $(SUBDIR64_$(MACH))

ROOT_MOD_DIR            = $(ROOT)/kernel

ROOT_KERN_DIR_32        = $(ROOT_MOD_DIR)
ROOT_BRAND_DIR_32       = $(ROOT_MOD_DIR)/brand
ROOT_DRV_DIR_32         = $(ROOT_MOD_DIR)/drv
ROOT_DTRACE_DIR_32      = $(ROOT_MOD_DIR)/dtrace
ROOT_EXEC_DIR_32        = $(ROOT_MOD_DIR)/exec
ROOT_FS_DIR_32          = $(ROOT_MOD_DIR)/fs
ROOT_SCHED_DIR_32       = $(ROOT_MOD_DIR)/sched
ROOT_SOCK_DIR_32        = $(ROOT_MOD_DIR)/socketmod
ROOT_STRMOD_DIR_32      = $(ROOT_MOD_DIR)/strmod
ROOT_IPP_DIR_32         = $(ROOT_MOD_DIR)/ipp
ROOT_SYS_DIR_32         = $(ROOT_MOD_DIR)/sys
ROOT_MISC_DIR_32        = $(ROOT_MOD_DIR)/misc
ROOT_KGSS_DIR_32        = $(ROOT_MOD_DIR)/misc/kgss
ROOT_SCSI_VHCI_DIR_32   = $(ROOT_MOD_DIR)/misc/scsi_vhci
ROOT_PMCS_FW_DIR_32     = $(ROOT_MOD_DIR)/misc/pmcs
ROOT_QLC_FW_DIR_32      = $(ROOT_MOD_DIR)/misc/qlc
ROOT_EMLXS_FW_DIR_32    = $(ROOT_MOD_DIR)/misc/emlxs
ROOT_NLMISC_DIR_32      = $(ROOT_MOD_DIR)/misc
ROOT_MACH_DIR_32        = $(ROOT_MOD_DIR)/mach
ROOT_CPU_DIR_32         = $(ROOT_MOD_DIR)/cpu
ROOT_TOD_DIR_32         = $(ROOT_MOD_DIR)/tod
ROOT_FONT_DIR_32        = $(ROOT_MOD_DIR)/fonts
ROOT_DACF_DIR_32        = $(ROOT_MOD_DIR)/dacf
ROOT_CRYPTO_DIR_32      = $(ROOT_MOD_DIR)/crypto
ROOT_MAC_DIR_32         = $(ROOT_MOD_DIR)/mac
ROOT_KICONV_DIR_32      = $(ROOT_MOD_DIR)/kiconv

ROOT_KERN_DIR_64        = $(ROOT_MOD_DIR)/$(SUBDIR64)
ROOT_BRAND_DIR_64       = $(ROOT_MOD_DIR)/brand/$(SUBDIR64)
ROOT_DRV_DIR_64         = $(ROOT_MOD_DIR)/drv/$(SUBDIR64)
ROOT_DTRACE_DIR_64      = $(ROOT_MOD_DIR)/dtrace/$(SUBDIR64)
ROOT_EXEC_DIR_64        = $(ROOT_MOD_DIR)/exec/$(SUBDIR64)
ROOT_FS_DIR_64          = $(ROOT_MOD_DIR)/fs/$(SUBDIR64)
ROOT_SCHED_DIR_64       = $(ROOT_MOD_DIR)/sched/$(SUBDIR64)
ROOT_SOCK_DIR_64        = $(ROOT_MOD_DIR)/socketmod/$(SUBDIR64)
ROOT_STRMOD_DIR_64      = $(ROOT_MOD_DIR)/strmod/$(SUBDIR64)
ROOT_IPP_DIR_64         = $(ROOT_MOD_DIR)/ipp/$(SUBDIR64)
ROOT_SYS_DIR_64         = $(ROOT_MOD_DIR)/sys/$(SUBDIR64)
ROOT_MISC_DIR_64        = $(ROOT_MOD_DIR)/misc/$(SUBDIR64)
ROOT_KGSS_DIR_64        = $(ROOT_MOD_DIR)/misc/kgss/$(SUBDIR64)
ROOT_SCSI_VHCI_DIR_64   = $(ROOT_MOD_DIR)/misc/scsi_vhci/$(SUBDIR64)
ROOT_PMCS_FW_DIR_64     = $(ROOT_MOD_DIR)/misc/pmcs/$(SUBDIR64)
ROOT_QLC_FW_DIR_64      = $(ROOT_MOD_DIR)/misc/qlc/$(SUBDIR64)
ROOT_EMLXS_FW_DIR_64    = $(ROOT_MOD_DIR)/misc/emlxs/$(SUBDIR64)
ROOT_NLMISC_DIR_64      = $(ROOT_MOD_DIR)/misc/$(SUBDIR64)
ROOT_MACH_DIR_64        = $(ROOT_MOD_DIR)/mach/$(SUBDIR64)
ROOT_CPU_DIR_64         = $(ROOT_MOD_DIR)/cpu/$(SUBDIR64)
ROOT_TOD_DIR_64         = $(ROOT_MOD_DIR)/tod/$(SUBDIR64)
ROOT_FONT_DIR_64        = $(ROOT_MOD_DIR)/fonts/$(SUBDIR64)
```

```
ROOT_DACF_DIR_64        = $(ROOT_MOD_DIR)/dacf/$(SUBDIR64)
ROOT_CRYPTO_DIR_64      = $(ROOT_MOD_DIR)/crypto/$(SUBDIR64)
ROOT_MAC_DIR_64         = $(ROOT_MOD_DIR)/mac/$(SUBDIR64)
ROOT_KICONV_DIR_64      = $(ROOT_MOD_DIR)/kiconv/$(SUBDIR64)

ROOT_KERN_DIR           = $(ROOT_KERN_DIR_$(CLASS))
ROOT_BRAND_DIR          = $(ROOT_BRAND_DIR_$(CLASS))
ROOT_DRV_DIR            = $(ROOT_DRV_DIR_$(CLASS))
ROOT_DTRACE_DIR         = $(ROOT_DTRACE_DIR_$(CLASS))
ROOT_EXEC_DIR           = $(ROOT_EXEC_DIR_$(CLASS))
ROOT_FS_DIR             = $(ROOT_FS_DIR_$(CLASS))
ROOT_SCHED_DIR          = $(ROOT_SCHED_DIR_$(CLASS))
ROOT_SOCK_DIR           = $(ROOT_SOCK_DIR_$(CLASS))
ROOT_STRMOD_DIR         = $(ROOT_STRMOD_DIR_$(CLASS))
ROOT_IPP_DIR            = $(ROOT_IPP_DIR_$(CLASS))
ROOT_SYS_DIR            = $(ROOT_SYS_DIR_$(CLASS))
ROOT_MISC_DIR           = $(ROOT_MISC_DIR_$(CLASS))
ROOT_KGSS_DIR           = $(ROOT_KGSS_DIR_$(CLASS))
ROOT_SCSI_VHCI_DIR      = $(ROOT_SCSI_VHCI_DIR_$(CLASS))
ROOT_PMCS_FW_DIR        = $(ROOT_PMCS_FW_DIR_$(CLASS))
ROOT_QLC_FW_DIR         = $(ROOT_QLC_FW_DIR_$(CLASS))
ROOT_EMLXS_FW_DIR       = $(ROOT_EMLXS_FW_DIR_$(CLASS))
ROOT_NLMISC_DIR         = $(ROOT_NLMISC_DIR_$(CLASS))
ROOT_MACH_DIR           = $(ROOT_MACH_DIR_$(CLASS))
ROOT_CPU_DIR            = $(ROOT_CPU_DIR_$(CLASS))
ROOT_TOD_DIR            = $(ROOT_TOD_DIR_$(CLASS))
ROOT_FONT_DIR           = $(ROOT_FONT_DIR_$(CLASS))
ROOT_DACF_DIR           = $(ROOT_DACF_DIR_$(CLASS))
ROOT_CRYPTO_DIR         = $(ROOT_CRYPTO_DIR_$(CLASS))
ROOT_MAC_DIR            = $(ROOT_MAC_DIR_$(CLASS))
ROOT_KICONV_DIR         = $(ROOT_KICONV_DIR_$(CLASS))
ROOT_FIRMWARE_DIR       = $(ROOT_MOD_DIR)/firmware

ROOT_MOD_DIRS_32        = $(ROOT_BRAND_DIR_32) $(ROOT_DRV_DIR_32)
ROOT_MOD_DIRS_32        = $(ROOT_BRAND_DIR_32) $(ROOT_DRV_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_EXEC_DIR_32) $(ROOT_DTRACE_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_FS_DIR_32) $(ROOT_SCHED_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_STRMOD_DIR_32) $(ROOT_SYS_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_IPP_DIR_32) $(ROOT_SOCK_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_MISC_DIR_32) $(ROOT_MACH_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_KGSS_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_SCSI_VHCI_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_PMCS_FW_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_QLC_FW_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_EMLXS_FW_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_CPU_DIR_32) $(ROOT_FONT_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_TOD_DIR_32) $(ROOT_DACF_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_CRYPTO_DIR_32) $(ROOT_MAC_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_KICONV_DIR_32)
ROOT_MOD_DIRS_32        += $(ROOT_FIRMWARE_DIR)

USR_MOD_DIR             = $(ROOT)/usr/kernel

USR_DRV_DIR_32          = $(USR_MOD_DIR)/drv
USR_EXEC_DIR_32         = $(USR_MOD_DIR)/exec
USR_FS_DIR_32           = $(USR_MOD_DIR)/fs
USR_SCHED_DIR_32        = $(USR_MOD_DIR)/sched
USR_SOCK_DIR_32         = $(USR_MOD_DIR)/socketmod
USR_STRMOD_DIR_32       = $(USR_MOD_DIR)/strmod
USR_SYS_DIR_32          = $(USR_MOD_DIR)/sys
USR_MISC_DIR_32         = $(USR_MOD_DIR)/misc
USR_DACF_DIR_32         = $(USR_MOD_DIR)/dacf
USR_PCBE_DIR_32         = $(USR_MOD_DIR)/pcbe
USR_DTRACE_DIR_32       = $(USR_MOD_DIR)/dtrace
USR_BRAND_DIR_32        = $(USR_MOD_DIR)/brand
```

```
USR_DRV_DIR_64          = $(USR_MOD_DIR)/drv/$(SUBDIR64)
USR_EXEC_DIR_64         = $(USR_MOD_DIR)/exec/$(SUBDIR64)
USR_FS_DIR_64           = $(USR_MOD_DIR)/fs/$(SUBDIR64)
USR_SCHED_DIR_64        = $(USR_MOD_DIR)/sched/$(SUBDIR64)
USR_SOCK_DIR_64         = $(USR_MOD_DIR)/socketmod/$(SUBDIR64)
USR_STRMOD_DIR_64       = $(USR_MOD_DIR)/strmod/$(SUBDIR64)
USR_SYS_DIR_64          = $(USR_MOD_DIR)/sys/$(SUBDIR64)
USR_MISC_DIR_64         = $(USR_MOD_DIR)/misc/$(SUBDIR64)
USR_DACF_DIR_64         = $(USR_MOD_DIR)/dacf/$(SUBDIR64)
USR_PCBE_DIR_64         = $(USR_MOD_DIR)/pcbe/$(SUBDIR64)
USR_DTRACE_DIR_64       = $(USR_MOD_DIR)/dtrace/$(SUBDIR64)
USR_BRAND_DIR_64        = $(USR_MOD_DIR)/brand/$(SUBDIR64)

USR_DRV_DIR             = $(USR_DRV_DIR_$(CLASS))
USR_EXEC_DIR            = $(USR_EXEC_DIR_$(CLASS))
USR_FS_DIR              = $(USR_FS_DIR_$(CLASS))
USR_SCHED_DIR           = $(USR_SCHED_DIR_$(CLASS))
USR_SOCK_DIR            = $(USR_SOCK_DIR_$(CLASS))
USR_STRMOD_DIR          = $(USR_STRMOD_DIR_$(CLASS))
USR_SYS_DIR             = $(USR_SYS_DIR_$(CLASS))
USR_MISC_DIR            = $(USR_MISC_DIR_$(CLASS))
USR_DACF_DIR            = $(USR_DACF_DIR_$(CLASS))
USR_PCBE_DIR            = $(USR_PCBE_DIR_$(CLASS))
USR_DTRACE_DIR          = $(USR_DTRACE_DIR_$(CLASS))
USR_BRAND_DIR           = $(USR_BRAND_DIR_$(CLASS))

USR_MOD_DIRS_32         = $(USR_DRV_DIR_32) $(USR_EXEC_DIR_32)
USR_MOD_DIRS_32         += $(USR_FS_DIR_32) $(USR_SCHED_DIR_32)
USR_MOD_DIRS_32         += $(USR_STRMOD_DIR_32) $(USR_SYS_DIR_32)
USR_MOD_DIRS_32         += $(USR_MISC_DIR_32) $(USR_DACF_DIR_32)
USR_MOD_DIRS_32         += $(USR_PCBE_DIR_32)
USR_MOD_DIRS_32         += $(USR_DTRACE_DIR_32) $(USR_BRAND_DIR_32)
USR_MOD_DIRS_32         += $(USR_SOCK_DIR_32)

#
#
#
include $(SRC)/Makefile.psm

#
#       The "-r" on the remove may be considered temporary, but is required
#       while the replacement of the SUNW,SPARCstation-10,SX directory by
#       a symbolic link is being propagated.
#
INS.slink1= $(RM) -r $@; $(SYMLINK) $(PLATFORM) $@
INS.slink2= $(RM) -r $@; $(SYMLINK) ../$(PLATFORM)/$(@F) $@
INS.slink3= $(RM) -r $@; $(SYMLINK) $(IMPLEMENTED_PLATFORM) $@
INS.slink4= $(RM) -r $@; $(SYMLINK) ../$(PLATFORM)/include $@
INS.slink5= $(RM) -r $@; $(SYMLINK) ../$(PLATFORM)/sbin $@
INS.slink6= $(RM) -r $@; $(SYMLINK) ../../$(PLATFORM)/lib/$(MODULE) $@
INS.slink7= $(RM) -r $@; $(SYMLINK) ../../$(PLATFORM)/sbin/$(@F) $@

ROOT_PLAT_LINKS          = $(PLAT_LINKS:%=$(ROOT_PLAT_DIR)/%)
ROOT_PLAT_LINKS_2        = $(PLAT_LINKS_2:%=$(ROOT_PLAT_DIR)/%)
USR_PLAT_LINKS           = $(PLAT_LINKS:%=$(USR_PLAT_DIR)/%)
USR_PLAT_LINKS_2         = $(PLAT_LINKS_2:%=$(USR_PLAT_DIR)/%)

#
# Collection of all relevant, delivered kernel modules.
#
# Note that we insist on building genunix first, because everything else
# uniquifies against it.  When doing a 'make' from usr/src/uts/, we'll enter
# the platform directories first.  These will cd into the corresponding genunix
# directory and build it.  So genunix /shouldn't/ get rebuilt when we get to
# building all the kernel modules.  However, due to an as-yet-unexplained
# problem with dependencies, sometimes it does get rebuilt, which then messes
```

```
# up the other modules.  So we always force the issue here rather than try to
# build genunix in parallel with everything else.
#
PARALLEL_KMODS = $(DRV_KMODS) $(EXEC_KMODS) $(FS_KMODS) $(SCHED_KMODS) \
                 $(TOD_KMODS) $(STRMOD_KMODS) $(SYS_KMODS) $(MISC_KMODS) \
                 $(NLMISC_KMODS) $(MACH_KMODS) $(CPU_KMODS) $(GSS_KMODS) \
                 $(MMU_KMODS) $(DACF_KMODS) $(EXPORT_KMODS) $(IPP_KMODS) \
                 $(CRYPTO_KMODS) $(PCBE_KMODS) \
                 $(DRV_KMODS_$(CLASS)) $(MISC_KMODS_$(CLASS)) $(MAC_KMODS) \
                 $(BRAND_KMODS) $(KICONV_KMODS) \
                 $(SOCKET_KMODS)

KMODS = $(GENUNIX_KMODS) $(PARALLEL_KMODS)

$(PARALLEL_KMODS): $(GENUNIX_KMODS)

LINT_KMODS = $(DRV_KMODS) $(EXEC_KMODS) $(FS_KMODS) $(SCHED_KMODS) \
          $(TOD_KMODS) $(STRMOD_KMODS) $(SYS_KMODS) $(MISC_KMODS) \
          $(MACH_KMODS) $(GSS_KMODS) $(DACF_KMODS) $(IPP_KMODS) \
          $(CRYPTO_KMODS) $(PCBE_KMODS) \
          $(DRV_KMODS_$(CLASS)) $(MISC_KMODS_$(CLASS)) $(MAC_KMODS) \
          $(BRAND_KMODS) $(KICONV_KMODS) $(SOCKET_KMODS)

#
#       Files to be compiled with -xa, to generate basic block execution
#       count data.
#
#       There are several ways to compile parts of the kernel for kcov:
#               1)  Add targets to BB_FILES here or in other Makefiles
#                       (they must in the form of $(OBJS_DIR)/target.o)
#               2)  setenv BB_FILES '$(XXX_OBJS:%=$(OBJS_DIR)/%)'
#               3)  setenv BB_FILES '$(OBJECTS)'
#
#       Do NOT setenv CFLAGS -xa, as that will cause infinite recursion
#       in unix_bb.o
#
BB_FILES =
$(BB_FILES)     := XAOPT = -xa

#
#       The idea here is for unix_bb.o to be in all kernels except the
#       kernel which actually gets shipped to customers.  In practice,
#       $(RELEASE_BUILD) is on for a number of the late beta and fcs builds.
#
$(NOT_RELEASE_BUILD)$(OBJS_DIR)/unix_bb.o   := CPPFLAGS     += -DKCOV
$(NOT_RELEASE_BUILD)$(OBJS_DIR)/unix_bb.ln  := CPPFLAGS     += -DKCOV

#
#       Do not let unix_bb.o get compiled with -xa!
#
$(OBJS_DIR)/unix_bb.o   := XAOPT =

#
# Privilege files
#
PRIVS_AWK = $(SRC)/uts/common/os/privs.awk
PRIVS_DEF = $(SRC)/uts/common/os/priv_defs

#
# USB device data
#
USBDEVS_AWK =   $(SRC)/uts/common/io/usb/usbdevs2h.awk
USBDEVS_DATA =  $(SRC)/uts/common/io/usb/usbdevs


#
```

```
# If we’re using the newer CTF tools, then we need to make sure that we
# are building with the private -X option to ctfconvert which allows us
# to fixup the struct cpu to account for machcpu.
#
$(BUILD_NEW_CTF_TOOLS)CTFCVTFLAGS += -X
```

```
**********************************************************
   13960 Mon Apr 15 12:20:22 2019
new/usr/src/uts/common/io/comstar/port/srpt/srpt_impl.h
10805 Fix for 10687 can be improved
**********************************************************
______unchanged_portion_omitted_

extern srpt_ctxt_t *srpt_ctxt;

/*
 * For Non recoverable or Major Errors
 */
#define SRPT_LOG_L0     0

/*
 * For additional information on Non recoverable errors and
 * warnings/informational message for sys-admin types.
 */
#define SRPT_LOG_L1     1

/*
 * debug only
 * for more verbose trace than L1, for e.g. recoverable errors,
 * or intersting trace
 */
#define SRPT_LOG_L2     2

/*
 * debug only
 * for more verbose trace than L2, for e.g. printing function entries....
 */
#define SRPT_LOG_L3     3

/*
 * debug only
 * for more verbose trace than L3, for e.g. printing minor function entries...
 */
#define SRPT_LOG_L4     4

/*
 * srpt_errlevel can be set in the debugger to enable additional logging.
 * You can also add set srpt:srpt_errlevel={0,1,2,3,4} in /etc/system.
 * The default log level is L1.
 */
#define SRPT_LOG_DEFAULT_LEVEL SRPT_LOG_L1

extern uint_t srpt_errlevel;


#define SRPT_DPRINTF_L0(...) cmn_err(CE_WARN, __VA_ARGS__)
#define SRPT_DPRINTF_L1(...) cmn_err(CE_NOTE, __VA_ARGS__)
#define SRPT_DPRINTF_L2(...)    if (srpt_errlevel >= SRPT_LOG_L2) { \
                                        cmn_err(CE_NOTE, __VA_ARGS__);\
                                }
#ifdef  DEBUG
#define SRPT_DPRINTF_L3(...)    if (srpt_errlevel >= SRPT_LOG_L3) { \
                                        cmn_err(CE_NOTE, __VA_ARGS__);\
                                }
#define SRPT_DPRINTF_L4(...)    if (srpt_errlevel >= SRPT_LOG_L4) { \
                                        cmn_err(CE_NOTE, __VA_ARGS__);\
                                }
#else
#define SRPT_DPRINTF_L3(...) (void)(0)
#define SRPT_DPRINTF_L4(...) (void)(0)
#define SRPT_DPRINTF_L3(...)
#define SRPT_DPRINTF_L4(...)
```

```
#endif

#ifdef __cplusplus
}
______unchanged_portion_omitted_
```

```
**********************************************************
    6526 Mon Apr 15 12:20:22 2019
new/usr/src/uts/common/io/pciex/pcieb.h
10805 Fix for 10687 can be improved
**********************************************************
/*
 * 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) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
 */

/*
 * Copyright 2019, Joyent, Inc.
 */

#ifndef _SYS_PCIEB_H
#define _SYS_PCIEB_H

#ifdef  __cplusplus
extern "C" {
#endif

#if defined(DEBUG)
#define PCIEB_DEBUG pcieb_dbg
extern void pcieb_dbg(uint_t bit, dev_info_t *dip, char *fmt, ...);
#else /* DEBUG */
#define PCIEB_DEBUG(...) (void)(0)
#define PCIEB_DEBUG(...)
#endif /* DEBUG */

typedef enum {  /* same sequence as pcieb_debug_sym[] */
        /*  0 */ DBG_ATTACH,
        /*  1 */ DBG_PWR,
        /*  2 */ DBG_INTR
} pcieb_debug_bit_t;
______unchanged_portion_omitted_
```

****
   29303 Mon Apr 15 12:20:22 2019
new/usr/src/uts/common/sys/ib/clients/eoib/eib_impl.h
10805 Fix for 10687 can be improved
****

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

/*
 * Copyright 2019, Joyent, Inc.
 */

#ifndef _SYS_IB_EOIB_EIB_IMPL_H
#define _SYS_IB_EOIB_EIB_IMPL_H

#ifdef __cplusplus
extern "C" {
#endif

#include <sys/ddi.h>
#include <sys/mac.h>
#include <sys/sunddi.h>
#include <sys/varargs.h>
#include <sys/vlan.h>
#include <sys/ib/ibtl/ibti.h>
#include <sys/ib/ibtl/ibvti.h>
#include <sys/ib/ib_pkt_hdrs.h>

#include <sys/ib/clients/eoib/fip.h>
#include <sys/ib/clients/eoib/eib.h>

/*
 * Driver specific constants
 */
#define EIB_E_SUCCESS                   0
#define EIB_E_FAILURE                   -1
#define EIB_MAX_LINE                    128
#define EIB_MAX_SGL                     59
#define EIB_MAX_POST_MULTIPLE           4
#define EIB_MAX_PAYLOAD_HDR_SZ          160
#define EIB_TX_COPY_THRESH              4096    /* greater than mtu */
#define EIB_MAX_VNICS                   64      /* do not change this */
#define EIB_LOGIN_TIMEOUT_USEC          8000000
#define EIB_RWR_CHUNK_SZ                8
#define EIB_IPHDR_ALIGN_ROOM            32
#define EIB_IP_HDR_ALIGN                2
#define EIB_MAX_RX_PKTS_ONINTR          0x800
#define EIB_MAX_LOGIN_ATTEMPTS          3
#define EIB_MAX_VHUB_TBL_ATTEMPTS       3
#define EIB_MAX_KA_ATTEMPTS             3
#define EIB_MAX_ATTEMPTS                10
#define EIB_DELAY_HALF_SECOND           500000
#define EIB_GRH_SZ                      (sizeof (ib_grh_t))

/*
 * Debug messages
 */
#define EIB_MSGS_CRIT           0x01
#define EIB_MSGS_ERR            0x02
#define EIB_MSGS_WARN           0x04
#define EIB_MSGS_DEBUG          0x08
#define EIB_MSGS_ARGS           0x10
#define EIB_MSGS_PKT            0x20
#define EIB_MSGS_VERBOSE        0x40
#define EIB_MSGS_DEFAULT        (EIB_MSGS_CRIT | EIB_MSGS_ERR | EIB_MSGS_WARN)

#define EIB_LOGSZ_DEFAULT       0x20000

#define EIB_DPRINTF_CRIT        eib_dprintf_crit
#define EIB_DPRINTF_ERR         eib_dprintf_err
#define EIB_DPRINTF_WARN        eib_dprintf_warn
#ifdef EIB_DEBUG
#define EIB_DPRINTF_DEBUG       eib_dprintf_debug
#define EIB_DPRINTF_ARGS        eib_dprintf_args
#define EIB_DPRINTF_PKT         eib_dprintf_pkt
#define EIB_DPRINTF_VERBOSE     eib_dprintf_verbose
#else
#define EIB_DPRINTF_DEBUG(...)  (void)(0)
#define EIB_DPRINTF_ARGS(...)   (void)(0)
#define EIB_DPRINTF_PKT(...)    (void)(0)
#define EIB_DPRINTF_VERBOSE(...) (void)(0)
#define EIB_DPRINTF_DEBUG(...)
#define EIB_DPRINTF_ARGS(...)
#define EIB_DPRINTF_PKT(...)
#define EIB_DPRINTF_VERBOSE(...)
#endif

/*
 *  EoIB threads to provide various services
 */
#define EIB_EVENTS_HDLR         "eib_events_handler"
#define EIB_RWQES_REFILLER      "eib_rwqes_refiller"
#define EIB_VNIC_CREATOR        "eib_vnic_creator"
#define EIB_TXWQES_MONITOR      "eib_txwqe_monitor"
#define EIB_LSOBUFS_MONITOR     "eib_lsobufs_monitor"

/*
 * Macro for finding the least significant bit set in a 64-bit unsigned int
 */
#define EIB_FIND_LSB_SET(val64) eib_setbit_mod67[((-(val64) & (val64)) % 67)]

/*
 * LSO buffers
 *
 * Under normal circumstances we should never need to use any buffer
 * that’s larger than MTU.  Unfortunately, IB HCA has limitations
 * on the length of SGL that are much smaller than those for regular
 * ethernet NICs.  Since the network layer doesn’t care to limit the
 * number of mblk fragments in any send mp chain, we end up having to
 * use these larger buffers occasionally.
 */
#define EIB_LSO_MAXLEN                  65536
#define EIB_LSO_BUFSZ                   8192
#define EIB_LSO_NUM_BUFS                1024
#define EIB_LSO_FREE_BUFS_THRESH        (EIB_LSO_NUM_BUFS >> 5)

typedef struct eib_lsobuf_s {
        struct eib_lsobuf_s *lb_next;
        uint8_t         *lb_buf;
        int             lb_isfree;
} eib_lsobuf_t;
______unchanged_portion_omitted_
```

```
**********************************************************
   14148 Mon Apr 15 12:20:22 2019
new/usr/src/uts/common/sys/ib/clients/eoib/enx_impl.h
10805 Fix for 10687 can be improved
**********************************************************
/*
 * 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) 2010, Oracle and/or its affiliates. All rights reserved.
 */

/*
 * Copyright 2019, Joyent, Inc.
 */

#ifndef _SYS_IB_EOIB_ENX_IMPL_H
#define _SYS_IB_EOIB_ENX_IMPL_H

#ifdef __cplusplus
extern "C" {
#endif

#include <sys/ddi.h>
#include <sys/sunddi.h>
#include <sys/varargs.h>
#include <sys/ib/ibtl/ibti.h>
#include <sys/ib/ibtl/ibvti.h>
#include <sys/ib/ib_pkt_hdrs.h>
#include <sys/ib/ibtl/impl/ibtl_ibnex.h>
#include <sys/ib/mgt/sm_attr.h>

#include <sys/ib/clients/eoib/fip.h>
#include <sys/ib/clients/eoib/eib.h>

/*
 * Driver specific constants
 */
#define ENX_E_SUCCESS           0
#define ENX_E_FAILURE           -1
#define ENX_MAX_LINE            128
#define ENX_GRH_SZ              (sizeof (ib_grh_t))

/*
 * Debug messages
 */
#define ENX_MSGS_CRIT           0x01
#define ENX_MSGS_ERR            0x02
```

```
#define ENX_MSGS_WARN           0x04
#define ENX_MSGS_DEBUG          0x08
#define ENX_MSGS_ARGS           0x10
#define ENX_MSGS_VERBOSE        0x20
#define ENX_MSGS_DEFAULT        (ENX_MSGS_CRIT | ENX_MSGS_ERR | ENX_MSGS_WARN)

#define ENX_LOGSZ_DEFAULT       0x20000

#define ENX_DPRINTF_CRIT        eibnx_dprintf_crit
#define ENX_DPRINTF_ERR         eibnx_dprintf_err
#define ENX_DPRINTF_WARN        eibnx_dprintf_warn
#ifdef ENX_DEBUG
#define ENX_DPRINTF_DEBUG       eibnx_dprintf_debug
#define ENX_DPRINTF_ARGS        eibnx_dprintf_args
#define ENX_DPRINTF_VERBOSE     eibnx_dprintf_verbose
#else
#define ENX_DPRINTF_DEBUG(...)  (void)(0)
#define ENX_DPRINTF_ARGS(...)   (void)(0)
#define ENX_DPRINTF_VERBOSE(...) (void)(0)
#define ENX_DPRINTF_DEBUG(...)
#define ENX_DPRINTF_ARGS(...)
#define ENX_DPRINTF_VERBOSE(...)
#endif

/*
 *  EoIB Nexus service threads
 */
#define ENX_PORT_MONITOR        "eibnx_port_%d_monitor"
#define ENX_NODE_CREATOR        "eibnx_node_creator"

/*
 * Default period (us) for unicast solicitations to discovered gateways.
 * EoIB specification requires that hosts send solicitation atleast every
 * 4 * GW_ADV_PERIOD.
 */
#define ENX_DFL_SOLICIT_PERIOD_USEC     32000000

/*
 * Portinfo list per HCA
 */
typedef struct eibnx_port_s {
        struct eibnx_port_s     *po_next;
        ibt_hca_portinfo_t      *po_pi;
        uint_t                  po_pi_size;
} eibnx_port_t;
______unchanged_portion_omitted_
```

```
**********************************************************
    2872 Mon Apr 15 12:20:23 2019
new/usr/src/uts/common/sys/ib/clients/rds/rdsib_debug.h
10805 Fix for 10687 can be improved
**********************************************************
/*
 * 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 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2019, Joyent, Inc.
 */

#ifndef _RDSIB_DEBUG_H
#define _RDSIB_DEBUG_H

#ifdef __cplusplus
extern "C" {
#endif

#define         LABEL   "RDS"

/*
 * warnings, console & syslog buffer.
 * For Non recoverable or Major Errors
 */
#define RDS_LOG_L0      0

/*
 * syslog buffer or RDS trace buffer (console if booted /w debug)
 * For additional information on Non recoverable errors and
 * warnings/informational message for sys-admin types.
 */
#define RDS_LOG_L1      1

/*
 * debug only
 * for more verbose trace than L1, for e.g. recoverable errors,
 * or intersting trace
 */
#define RDS_LOG_L2      2

/*
 * debug only
 * for more verbose trace than L2, for e.g. informational messages
```

```
 */
#define RDS_LOG_L3      3

/*
 * debug only
 * for more verbose trace than L3, for e.g. printing function entries...
 */
#define RDS_LOG_L4      4

/*
 * debug only
 * most verbose level. Used only for  excessive trace, for e.g.
 * printing structures etc.
 */
#define RDS_LOG_L5      5

/*
 * debug only
 * for messages from softints, taskqs, intr handlers, timeout handlers etc.
 */
#define RDS_LOG_LINTR   6


#ifdef DEBUG
#define RDS_DPRINTF_INTR        rds_dprintf_intr
#define RDS_DPRINTF5            rds_dprintf5
#define RDS_DPRINTF4            rds_dprintf4
#define RDS_DPRINTF3            rds_dprintf3

void rds_dprintf_intr(
                char            *name,
                char            *fmt, ...);
void rds_dprintf5(
                char            *name,
                char            *fmt, ...);
void rds_dprintf4(
                char            *name,
                char            *fmt, ...);
void rds_dprintf3(
                char            *name,
                char            *fmt, ...);
#else
#define RDS_DPRINTF_INTR(...)   (void)(0)
#define RDS_DPRINTF5(...)       (void)(0)
#define RDS_DPRINTF4(...)       (void)(0)
#define RDS_DPRINTF3(...)       (void)(0)
#define RDS_DPRINTF_INTR(...)
#define RDS_DPRINTF5(...)
#define RDS_DPRINTF4(...)
#define RDS_DPRINTF3(...)
#endif

#define RDS_DPRINTF2    rds_dprintf2
#define RDS_DPRINTF1    rds_dprintf1
#define RDS_DPRINTF0    rds_dprintf0

void rds_dprintf2(
                char            *name,
                char            *fmt, ...);
void rds_dprintf1(
                char            *name,
                char            *fmt, ...);
void rds_dprintf0(
                char            *name,
                char            *fmt, ...);
```

```
#ifdef __cplusplus
}
______unchanged_portion_omitted_
```

```
**********************************************************
    3204 Mon Apr 15 12:20:23 2019
new/usr/src/uts/common/sys/ib/clients/rdsv3/rdsv3_debug.h
10805 Fix for 10687 can be improved
**********************************************************
```

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

/*
 * Copyright 2019, Joyent, Inc.
 */

#ifndef _RDSV3_DEBUG_H
#define _RDSV3_DEBUG_H

#ifdef __cplusplus
extern "C" {
#endif

#define         LABEL   "RDSV3"

/*
 * warnings, console & syslog buffer.
 * For Non recoverable or Major Errors
 */
#define RDSV3_LOG_L0    0

/*
 * syslog buffer or RDS trace buffer (console if booted /w debug)
 * For additional information on Non recoverable errors and
 * warnings/informational message for sys-admin types.
 */
#define RDSV3_LOG_L1    1

/*
 * debug only
 * for more verbose trace than L1, for e.g. recoverable errors,
 * or intersting trace
 */
#define RDSV3_LOG_L2    2

/*
 * debug only
 * for more verbose trace than L2, for e.g. informational messages
 */
```

```
#define RDSV3_LOG_L3    3

/*
 * debug only
 * for more verbose trace than L3, for e.g. printing function entries...
 */
#define RDSV3_LOG_L4    4

/*
 * debug only
 * most verbose level. Used only for  excessive trace, for e.g.
 * printing structures etc.
 */
#define RDSV3_LOG_L5    5

/*
 * debug only
 * for messages from softints, taskqs, intr handlers, timeout handlers etc.
 */
#define RDSV3_LOG_LINTR 6


#ifdef DEBUG
#define RDSV3_DPRINTF_INTR      rdsv3_dprintf_intr
#define RDSV3_DPRINTF5          rdsv3_dprintf5
#define RDSV3_DPRINTF4          rdsv3_dprintf4
#define RDSV3_DPRINTF3          rdsv3_dprintf3

void rdsv3_dprintf_intr(
                char            *name,
                char            *fmt, ...);
void rdsv3_dprintf5(
                char            *name,
                char            *fmt, ...);
void rdsv3_dprintf4(
                char            *name,
                char            *fmt, ...);
void rdsv3_dprintf3(
                char            *name,
                char            *fmt, ...);
#else
#define RDSV3_DPRINTF_INTR(...) (void)(0)
#define RDSV3_DPRINTF5(...)     (void)(0)
#define RDSV3_DPRINTF4(...)     (void)(0)
#define RDSV3_DPRINTF3(...)     (void)(0)
#define RDSV3_DPRINTF_INTR(...)
#define RDSV3_DPRINTF5(...)
#define RDSV3_DPRINTF4(...)
#define RDSV3_DPRINTF3(...)
#endif

#define RDSV3_DPRINTF2          rdsv3_dprintf2
#define RDSV3_DPRINTF1          rdsv3_dprintf1
#define RDSV3_DPRINTF0          rdsv3_dprintf0

void rdsv3_dprintf2(
                char            *name,
                char            *fmt, ...);
void rdsv3_dprintf1(
                char            *name,
                char            *fmt, ...);
void rdsv3_dprintf0(
                char            *name,
                char            *fmt, ...);

void rdsv3_trace(
                char            *name,
                uint8_t         lvl,
                char            *fmt, ...);

void rdsv3_vprintk(
                char            *name,
                uint8_t         lvl,
                const char      *fmt,
                va_list         ap);

/* defined in rds_debug.c */
void rdsv3_logging_initialization();
void rdsv3_logging_destroy();
int rdsv3_printk_ratelimit(void);

#ifdef __cplusplus
}
```

______***unchanged_portion_omitted_***

****
    3745 Mon Apr 15 12:20:23 2019
new/usr/src/uts/common/sys/ib/ibtl/impl/ibtl_util.h
10805 Fix for 10687 can be improved
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License, Version 1.0 only
 * (the "License").  You may not use this file except in compliance
 * with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2019, Joyent, Inc.
 */

#ifndef _SYS_IB_IBTL_IMPL_IBTL_UTIL_H
#define _SYS_IB_IBTL_IMPL_IBTL_UTIL_H

/*
 * ibtl_util.h
 *
 * All data structures and function prototypes that serve as helper
 * routines for IBTF implementation.
 */

#ifdef __cplusplus
extern "C" {
#endif

#include <sys/ib/ib_types.h>
#include <sys/varargs.h>

/*
 * Time Related Functions
 *
 *   ibt_usec2ib
 *      This function converts the standard input time in microseconds to
 *      IB’s 6 bits of timeout exponent, calculated based on
 *      time = 4.096us * 2 ^ exp.
 *
 *   ibt_ib2usec
 *      This function converts the input IB timeout exponent (6 bits) to
 *      standard time in microseconds, calculated based on
 *      time = 4.096us * 2 ^ exp.
 */
ib_time_t       ibt_usec2ib(clock_t microsecs);
```

```
clock_t         ibt_ib2usec(ib_time_t ib_time);


/*
 * IB logging, debug and console message handling
 */


/*
 * warnings, console & syslog buffer.
 * For Non recoverable or Major Errors
 */
#define IBTF_LOG_L0     0

/*
 * syslog buffer or IBTF trace buffer (console if booted /w debug)
 * For additional information on Non recoverable errors and
 * warnings/informational message for sys-admin types.
 */
#define IBTF_LOG_L1     1

/*
 * debug only
 * for more verbose trace than L1, for e.g. recoverable errors,
 * or intersting trace
 */
#define IBTF_LOG_L2     2

/*
 * debug only
 * for more verbose trace than L2, for e.g. printing function entries....
 */
#define IBTF_LOG_L3     3

/*
 * debug only
 * for more verbose trace than L3, for e.g. printing minor function entries...
 */
#define IBTF_LOG_L4     4

/*
 * debug only
 * most verbose level. Used only for  excessive trace, for e.g.
 * printing structures etc.
 */
#define IBTF_LOG_L5     5

/*
 * debug only
 * for messages from softints, taskqs, intr handlers, timeout handlers etc.
 * Only gets printed if "ibtl_allow_intr_msgs" is set
 */
#define IBTF_LOG_LINTR  6


#ifdef DEBUG
#define IBTF_DPRINTF_LINTR      ibtl_dprintf_intr
#define IBTF_DPRINTF_L5         ibtl_dprintf5
#define IBTF_DPRINTF_L4         ibtl_dprintf4
#define IBTF_DPRINTF_L3         ibtl_dprintf3

void ibtl_dprintf_intr(
                char            *name,
                char            *fmt, ...);
void ibtl_dprintf5(
                char            *name,
                char            *fmt, ...);
void ibtl_dprintf4(
                char            *name,
                char            *fmt, ...);
void ibtl_dprintf3(
                char            *name,
                char            *fmt, ...);
#else
#define IBTF_DPRINTF_LINTR(...) (void)(0)
#define IBTF_DPRINTF_L5(...)    (void)(0)
#define IBTF_DPRINTF_L4(...)    (void)(0)
#define IBTF_DPRINTF_L3(...)    (void)(0)
#define IBTF_DPRINTF_LINTR(...)
#define IBTF_DPRINTF_L5(...)
#define IBTF_DPRINTF_L4(...)
#define IBTF_DPRINTF_L3(...)
#endif

#define IBTF_DPRINTF_L2 ibtl_dprintf2
#define IBTF_DPRINTF_L1 ibtl_dprintf1
#define IBTF_DPRINTF_L0 ibtl_dprintf0

void ibtl_dprintf2(
                char            *name,
                char            *fmt, ...);
void ibtl_dprintf1(
                char            *name,
                char            *fmt, ...);
void ibtl_dprintf0(
                char            *name,
                char            *fmt, ...);

#ifdef __cplusplus
}
______unchanged_portion_omitted_
```

```
**********************************************************
   24339 Mon Apr 15 12:20:23 2019
new/usr/src/uts/common/sys/pcie_impl.h
10805 Fix for 10687 can be improved
**********************************************************
______unchanged_portion_omitted_

/*
 * Default interrupt priority for all PCI and PCIe nexus drivers including
 * hotplug interrupts.
 */
#define PCIE_INTR_PRI           (LOCK_LEVEL - 1)

/*
 * XXX - PCIE_IS_PCIE check is required in order not to invoke these macros
 * for non-standard PCI or PCI Express Hotplug Controllers.
 */
#define PCIE_ENABLE_ERRORS(dip) \
        if (PCIE_IS_PCIE(PCIE_DIP2BUS(dip))) {  \
                pcie_enable_errors(dip);        \
                (void) pcie_enable_ce(dip);     \
        }

#define PCIE_DISABLE_ERRORS(dip)                \
        if (PCIE_IS_PCIE(PCIE_DIP2BUS(dip))) {  \
                pcie_disable_errors(dip);       \
        }

/*
 * pcie_init_buspcie_fini_bus specific flags
 */
#define PCIE_BUS_INITIAL        0x0001
#define PCIE_BUS_FINAL          0x0002
#define PCIE_BUS_ALL            (PCIE_BUS_INITIAL | PCIE_BUS_FINAL)

#ifdef  DEBUG
#define PCIE_DBG pcie_dbg
/* Common Debugging shortcuts */
#define PCIE_DBG_CFG(dip, bus_p, name, sz, off, org) \
        PCIE_DBG("%s:%d:(0x%x) %s(0x%x) 0x%x -> 0x%x\n", ddi_node_name(dip), \
            ddi_get_instance(dip), bus_p->bus_bdf, name, off, org, \
            PCIE_GET(sz, bus_p, off))
#define PCIE_DBG_CAP(dip, bus_p, name, sz, off, org) \
        PCIE_DBG("%s:%d:(0x%x) %s(0x%x) 0x%x -> 0x%x\n", ddi_node_name(dip), \
            ddi_get_instance(dip), bus_p->bus_bdf, name, off, org, \
            PCIE_CAP_GET(sz, bus_p, off))
#define PCIE_DBG_AER(dip, bus_p, name, sz, off, org) \
        PCIE_DBG("%s:%d:(0x%x) %s(0x%x) 0x%x -> 0x%x\n", ddi_node_name(dip), \
            ddi_get_instance(dip), bus_p->bus_bdf, name, off, org, \
            PCIE_AER_GET(sz, bus_p, off))

#else   /* DEBUG */

#define PCIE_DBG_CFG(...)       (void)(0)
#define PCIE_DBG(...)           (void)(0)
#define PCIE_ARI_DBG(...)       (void)(0)
#define PCIE_DBG_CAP(...)       (void)(0)
#define PCIE_DBG_AER(...)       (void)(0)
#define PCIE_DBG_CFG(...)
#define PCIE_DBG(...)
#define PCIE_ARI_DBG(...)
#define PCIE_DBG_CAP(...)
#define PCIE_DBG_AER(...)

#endif  /* DEBUG */
```

```
/* PCIe Friendly Functions */
extern int pcie_init(dev_info_t *dip, caddr_t arg);
extern int pcie_uninit(dev_info_t *dip);
extern int pcie_hpintr_enable(dev_info_t *dip);
extern int pcie_hpintr_disable(dev_info_t *dip);
extern int pcie_intr(dev_info_t *dip);
extern int pcie_open(dev_info_t *dip, dev_t *devp, int flags, int otyp,
    cred_t *credp);
extern int pcie_close(dev_info_t *dip, dev_t dev, int flags, int otyp,
    cred_t *credp);
extern int pcie_ioctl(dev_info_t *dip, dev_t dev, int cmd, intptr_t arg,
    int mode, cred_t *credp, int *rvalp);
extern int pcie_prop_op(dev_t dev, dev_info_t *dip, ddi_prop_op_t prop_op,
    int flags, char *name, caddr_t valuep, int *lengthp);

extern void pcie_init_root_port_mps(dev_info_t *dip);
extern int pcie_initchild(dev_info_t *dip);
extern void pcie_uninitchild(dev_info_t *dip);
extern int pcie_init_cfghdl(dev_info_t *dip);
extern void pcie_fini_cfghdl(dev_info_t *dip);
extern void pcie_clear_errors(dev_info_t *dip);
extern int pcie_postattach_child(dev_info_t *dip);
extern void pcie_enable_errors(dev_info_t *dip);
extern void pcie_disable_errors(dev_info_t *dip);
extern int pcie_enable_ce(dev_info_t *dip);
extern boolean_t pcie_bridge_is_link_disabled(dev_info_t *);

extern pcie_bus_t *pcie_init_bus(dev_info_t *dip, pcie_req_id_t bdf,
    uint8_t flags);
extern void pcie_fini_bus(dev_info_t *dip, uint8_t flags);
extern void pcie_fab_init_bus(dev_info_t *dip, uint8_t flags);
extern void pcie_fab_fini_bus(dev_info_t *dip, uint8_t flags);
extern void pcie_rc_init_bus(dev_info_t *dip);
extern void pcie_rc_fini_bus(dev_info_t *dip);
extern void pcie_rc_init_pfd(dev_info_t *dip, pf_data_t *pfd);
extern void pcie_rc_fini_pfd(pf_data_t *pfd);
extern boolean_t pcie_is_child(dev_info_t *dip, dev_info_t *rdip);
extern int pcie_get_bdf_from_dip(dev_info_t *dip, pcie_req_id_t *bdf);
extern dev_info_t *pcie_get_my_childs_dip(dev_info_t *dip, dev_info_t *rdip);
extern uint32_t pcie_get_bdf_for_dma_xfer(dev_info_t *dip, dev_info_t *rdip);
extern int pcie_dev(dev_info_t *dip);
extern void pcie_get_fabric_mps(dev_info_t *rc_dip, dev_info_t *dip,
        int *max_supported);
extern int pcie_root_port(dev_info_t *dip);
extern int pcie_initchild_mps(dev_info_t *dip);
extern void pcie_set_rber_fatal(dev_info_t *dip, boolean_t val);
extern boolean_t pcie_get_rber_fatal(dev_info_t *dip);

extern uint32_t pcie_get_aer_uce_mask();
extern uint32_t pcie_get_aer_ce_mask();
extern uint32_t pcie_get_aer_suce_mask();
extern uint32_t pcie_get_serr_mask();
extern void pcie_set_aer_uce_mask(uint32_t mask);
extern void pcie_set_aer_ce_mask(uint32_t mask);
extern void pcie_set_aer_suce_mask(uint32_t mask);
extern void pcie_set_serr_mask(uint32_t mask);
extern void pcie_init_plat(dev_info_t *dip);
extern void pcie_fini_plat(dev_info_t *dip);
extern int pcie_read_only_probe(dev_info_t *, char *, dev_info_t **);
extern dev_info_t *pcie_func_to_dip(dev_info_t *dip, pcie_req_id_t function);
extern int pcie_ari_disable(dev_info_t *dip);
extern int pcie_ari_enable(dev_info_t *dip);

#define PCIE_ARI_FORW_NOT_SUPPORTED     0
#define PCIE_ARI_FORW_SUPPORTED         1
```

```
extern int pcie_ari_supported(dev_info_t *dip);

#define PCIE_ARI_FORW_DISABLED  0
#define PCIE_ARI_FORW_ENABLED   1

extern int pcie_ari_is_enabled(dev_info_t *dip);

#define PCIE_NOT_ARI_DEVICE             0
#define PCIE_ARI_DEVICE                 1

extern int pcie_ari_device(dev_info_t *dip);
extern int pcie_ari_get_next_function(dev_info_t *dip, int *func);

/* PCIe error handling functions */
extern void pf_eh_enter(pcie_bus_t *bus_p);
extern void pf_eh_exit(pcie_bus_t *bus_p);
extern int pf_scan_fabric(dev_info_t *rpdip, ddi_fm_error_t *derr,
    pf_data_t *root_pfd_p);
extern void pf_init(dev_info_t *, ddi_iblock_cookie_t, ddi_attach_cmd_t);
extern void pf_fini(dev_info_t *, ddi_detach_cmd_t);
extern int pf_hdl_lookup(dev_info_t *, uint64_t, uint32_t, uint64_t,
    pcie_req_id_t);
extern int pf_tlp_decode(pcie_bus_t *, pf_pcie_adv_err_regs_t *);
extern void pcie_force_fullscan();

#ifdef  DEBUG
extern uint_t pcie_debug_flags;
extern void pcie_dbg(char *fmt, ...);
#endif  /* DEBUG */

/* PCIe IOV functions */
extern dev_info_t *pcie_find_dip_by_bdf(dev_info_t *rootp, pcie_req_id_t bdf);

extern boolean_t pf_in_bus_range(pcie_bus_t *, pcie_req_id_t);
extern boolean_t pf_in_assigned_addr(pcie_bus_t *, uint64_t);
extern int pf_pci_decode(pf_data_t *, uint16_t *);
extern pcie_bus_t *pf_find_busp_by_bdf(pf_impl_t *, pcie_req_id_t);
extern pcie_bus_t *pf_find_busp_by_addr(pf_impl_t *, uint64_t);
extern pcie_bus_t *pf_find_busp_by_aer(pf_impl_t *, pf_data_t *);
extern pcie_bus_t *pf_find_busp_by_saer(pf_impl_t *, pf_data_t *);

extern int pciev_eh(pf_data_t *, pf_impl_t *);
extern pcie_bus_t *pciev_get_affected_dev(pf_impl_t *, pf_data_t *,
    uint16_t, uint16_t);
extern void pciev_eh_exit(pf_data_t *, uint_t);
extern boolean_t pcie_in_domain(pcie_bus_t *, uint_t);

#define PCIE_ZALLOC(data) kmem_zalloc(sizeof (data), KM_SLEEP)


#ifdef  __cplusplus
}
```

______unchanged_portion_omitted_

```
**********************************************************
   25821 Mon Apr 15 12:20:23 2019
new/usr/src/uts/common/sys/usb/scsa2usb/scsa2usb.h
10805 Fix for 10687 can be improved
**********************************************************
______unchanged_portion_omitted_

#define SCSA2USB_MK_32BIT(a, b, c, d) \
                (((a) << 24) | ((b) << 16) | ((c) << 8) | (d))

/* position of fields for SCMD_READ_CD CDB */
#define SCSA2USB_READ_CD_LEN_0  6       /* LEN[0] of SCMD_READ_CD */
#define SCSA2USB_READ_CD_LEN_1  7       /* LEN[1] of SCMD_READ_CD */
#define SCSA2USB_READ_CD_LEN_2  8       /* LEN[2] of SCMD_READ_CD */

/* macro to calculate LEN for SCMD_READ_CD command */
#define SCSA2USB_LEN_READ_CD(pkt) \
        (((pkt)->pkt_cdbp[SCSA2USB_READ_CD_LEN_0] << 16) +\
            ((pkt)->pkt_cdbp[SCSA2USB_READ_CD_LEN_1] << 8) +\
            (pkt)->pkt_cdbp[SCSA2USB_READ_CD_LEN_2])

/* Figure out Block Size before issuing a WRITE to CD-RW device */
#define SCSA2USB_CDRW_BLKSZ(bcount, len)        ((bcount) / (len));
#define SCSA2USB_VALID_CDRW_BLKSZ(blksz) \
        (((blksz) == CDROM_BLK_2048) || ((blksz) == CDROM_BLK_2352) || \
        ((blksz) == CDROM_BLK_2336) || ((blksz) == CDROM_BLK_2324) || \
        ((blksz) == 0))

/* debug and error msg logging */
#define DPRINT_MASK_SCSA        0x0001          /* for SCSA */
#define DPRINT_MASK_ATTA        0x0002          /* for ATTA */
#define DPRINT_MASK_EVENTS      0x0004          /* for event handling */
#define DPRINT_MASK_CALLBACKS   0x0008          /* for callbacks  */
#define DPRINT_MASK_TIMEOUT     0x0010          /* for timeouts */
#define DPRINT_MASK_DUMPING     0x0020          /* for dumping */
#define DPRINT_MASK_PM          0x0040          /* for pwr mgmt */
#define DPRINT_MASK_ALL         0xffffffff      /* for everything */

#ifdef  DEBUG
#define SCSA2USB_PRINT_CDB      scsa2usb_print_cdb
#else
#define SCSA2USB_PRINT_CDB(...) (void)(0)
#define SCSA2USB_PRINT_CDB(...)
#endif

/* ugen support */
#define SCSA2USB_MINOR_UGEN_BITS_MASK   0xff
#define SCSA2USB_MINOR_INSTANCE_MASK    ~SCSA2USB_MINOR_UGEN_BITS_MASK
#define SCSA2USB_MINOR_INSTANCE_SHIFT   8

#define SCSA2USB_MINOR_TO_INSTANCE(minor)       \
                (((minor) & SCSA2USB_MINOR_INSTANCE_MASK) >> \
                SCSA2USB_MINOR_INSTANCE_SHIFT)

#ifdef __cplusplus
}
______unchanged_portion_omitted_
```