```
**********************************************************
   27226 Wed Feb 26 14:23:23 2014
new/usr/src/man/man9f/Intro.9f
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
   1 '\" te
   2 .\" Copyright 2014 Pluribus Networks, Inc.
   3 .\" Copyright 2012 Garrett D'Amore <garrett@damore.org>.
   2 .\" Copyright 2012 Garrett D'Amore <garrett@damore.org>.  All rights reserved.
   4 .\" Copyright (c) 2005, Sun Microsystems, Inc., All Rights Reserved
   5 .\" The contents of this file are subject to the terms of the Common Development
   6 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
   7 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
   8 .TH INTRO 9F "Feb 26, 2014"
   7 .TH INTRO 9F "Feb 06, 2012"
   9 .SH NAME
  10 Intro, intro \- introduction to DDI/DKI functions
  11 .SH DESCRIPTION
  12 .sp
  13 .LP
  14 Section 9F describes the kernel functions available for use by device drivers.
  15 See \fBIntro\fR(9E) for an overview of device driver interfaces.
  16 .sp
  17 .LP
  18 In this section, the information for each driver function is organized under
  19 the following headings:
  20 .RS +4
  21 .TP
  22 .ie t \(bu
  23 .el o
  24 \fBNAME\fR summarizes the function's purpose.
  25 .RE
  26 .RS +4
  27 .TP
  28 .ie t \(bu
  29 .el o
  30 \fBSYNOPSIS\fR shows the syntax of the function's entry point in the source
  31 code. \fB#include\fR directives are shown for required headers.
  32 .RE
  33 .RS +4
  34 .TP
  35 .ie t \(bu
  36 .el o
  37 \fBINTERFACE\fR \fBLEVEL\fR describes any architecture dependencies.
  38 .RE
  39 .RS +4
  40 .TP
  41 .ie t \(bu
  42 .el o
  43 \fBARGUMENTS\fR describes any arguments required to invoke the function.
  44 .RE
  45 .RS +4
  46 .TP
  47 .ie t \(bu
  48 .el o
  49 \fBDESCRIPTION\fR describes general information about the function.
  50 .RE
  51 .RS +4
  52 .TP
  53 .ie t \(bu
  54 .el o
  55 \fBRETURN\fR \fBVALUES\fR describes the return values and messages that can
  56 result from invoking the function.
  57 .RE
  58 .RS +4
```



```
  59 .TP
  60 .ie t \(bu
  61 .el o
  62 \fBCONTEXT\fR indicates from which driver context (user, kernel, interrupt, or
  63 high-level interrupt) the function can be called.
  64 .RE
  65 .RS +4
  66 .TP
  67 .ie t \(bu
  68 .el o
  69 A driver function has \fIuser context\fR if it was directly invoked because of
  70 a user thread. The \fBread\fR(9E) entry point of the driver, invoked by a
  71 \fBread\fR(2) system call, has user context.
  72 .RE
  73 .RS +4
  74 .TP
  75 .ie t \(bu
  76 .el o
  77 A driver function has \fIkernel context\fR if was invoked by some other part of
  78 the kernel. In a block device driver, the \fBstrategy\fR(9E) entry point may be
  79 called by the page daemon to write pages to the device. The page daemon has no
  80 relation to the current user thread, so in this case \fBstrategy\fR(9E) has
  81 kernel context.
  82 .RE
  83 .RS +4
  84 .TP
  85 .ie t \(bu
  86 .el o
  87 \fIInterrupt context\fR is kernel context, but also has an interrupt level
  88 associated with it. Driver interrupt routines have interrupt context.
  89 .sp
  90 Note that a mutex acquired in user or kernel context that can also be acquired
  91 in interrupt context means that the user or kernel context thread holding that
  92 mutex is subject to all the restrictions imposed by interrupt context, for the
  93 duration of the ownership of that mutex. Please see the \fBmutex\fR(9F) man
  94 page for a more complete discussion of proper mutex handling for drivers.
  95 .RE
  96 .RS +4
  97 .TP
  98 .ie t \(bu
  99 .el o
 100 \fIHigh-level interrupt context\fR is a more restricted form of interrupt
 101 context. If a driver interrupt priority returned from
 102 \fBddi_intr_get_pri\fR(9F) is greater than the priority returned from
 103 \fBddi_intr_get_hilevel_pri\fR(9F) this indicates the interrupt handler will
 104 run in high-level interrupt context. These interrupt routines are only allowed
 105 to call \fBddi_intr_trigger_softint\fR(9F), \fBmutex_enter\fR(9F), and
 106 \fBmutex_exit\fR(9F). Furthermore, \fBmutex_enter\fR(9F) and
 107 \fBmutex_exit\fR(9F) may only be called on mutexes initialized with the
 108 interrupt priority returned by \fBddi_intr_get_pri\fR(9F).
 109 .RE
 110 .RS +4
 111 .TP
 112 .ie t \(bu
 113 .el o
 114 \fBSEE ALSO\fR indicates functions that are related by usage and sources, and
 115 which can be referred to for further information.
 116 .RE
 117 .RS +4
 118 .TP
 119 .ie t \(bu
 120 .el o
 121 \fBEXAMPLES\fR shows how the function can be used in driver code.
 122 .RE
 123 .sp
 124 .LP
```

```
125 Every driver MUST include <\fBsys/ddi.h\fR> and <\fBsys/sunddi.h\fR>, in that
126 order, and as the last files the driver includes.
127 .SH STREAMS KERNEL FUNCTION SUMMARY
128 .sp
129 .LP
130 The following table summarizes the STREAMS functions described in this section.
131 .sp

133 .sp
134 .TS
135 c c
136 l l .
137 Routine Type
138 _
139 \fBadjmsg\fR    DDI/DKI
140 \fBallocb\fR    DDI/DKI
141 \fBallocb_tmpl\fR       Solaris DDI
142 \fBbackq\fR     DDI/DKI
143 \fBbcanput\fR   DDI/DKI
144 \fBbcanputnext\fR       DDI/DKI
145 \fBbufcall\fR   DDI/DKI
146 \fBcanput\fR    DDI/DKI
147 \fBcanputnext\fR        DDI/DKI
148 \fBclrbuf\fR    DDI/DKI
149 \fBcopyb\fR     DDI/DKI
150 \fBcopymsg\fR   DDI/DKI
151 \fBDB_BASE\fR   Solaris DDI
152 \fBDB_LIM\fR    Solaris DDI
153 \fBDB_REF\fR    Solaris DDI
154 \fBDB_TYPE\fR   Solaris DDI
155 \fBdatamsg\fR   DDI/DKI
156 \fBdupb\fR      DDI/DKI
157 \fBdupmsg\fR    DDI/DKI
158 \fBenableok\fR  DDI/DKI
159 \fBesballoc\fR  DDI/DKI
160 \fBesbbcall\fR  DDI/DKI
161 \fBflushband\fR DDI/DKI
162 \fBflushq\fR    DDI/DKI
163 \fBfreeb\fR     DDI/DKI
164 \fBfreemsg\fR   DDI/DKI
165 \fBfreezestr\fR DDI/DKI
166 \fBgetq\fR      DDI/DKI
167 \fBIOC_CONVER_FROM\fR   Solaris DDI
168 \fBinsq\fR      DDI/DKI
169 \fBlinkb\fR     DDI/DKI
170 \fBMBLKHEAD\fR  Solaris DDI
171 \fBMBLKIN\fR    Solaris DDI
172 \fBMBLKL\fR     Solaris DDI
173 \fBMBLKSIZE\fR  Solaris DDI
174 \fBMBLKTAIL\fR  Solaris DDI
175 \fBmcopyin\fR   Solaris DDI
176 \fBmcopymsg\fR  Solaris DDI
177 \fBmcopyout\fR  Solaris DDI
178 \fBmerror\fR    Solaris DDI
179 \fBmexchange\fR Solaris DDI
180 \fBmioc2ack\fR  Solaris DDI
181 \fBmiocack\fR   Solaris DDI
182 \fBmexchange\fR Solaris DDI
183 \fBmiocpullup\fR        Solaris DDI
184 \fBmkiocb\fR    Solaris DDI
185 \fBmsgdsize\fR  DDI/DKI
186 \fBmsgpullup\fR DDI/DKI
187 \fBmsgsize\fR   Solaris DDI
188 \fBmt-streams\fR        Solaris DDI
189 \fBnoenable\fR  DDI/DKI
190 \fBOTHERQ\fR    DDI/DKI
```



```
191 \fBpullupmsg\fR DDI/DKI
192 \fBput\fR       DDI/DKI
193 \fBputbq\fR     DDI/DKI
194 \fBputctl\fR    DDI/DKI
195 \fBputctl1\fR   DDI/DKI
196 \fBputnext\fR   DDI/DKI
197 \fBputnextctl\fR        DDI/DKI
198 \fBputq\fR      DDI/DKI
199 \fBqassociate\fR        Solaris DDI
200 \fBqbufcall\fR  Solaris DDI
201 \fBqenable\fR   DDI/DKI
202 \fBqprocson\fR  DDI/DKI
203 \fBqprocsoff\fR DDI/DKI
204 \fBqreply\fR    DDI/DKI
205 \fBqsize\fR     DDI/DKI
206 \fBqtimeout\fR  Solaris DDI
207 \fBqunbufcall\fR        Solaris DDI
208 \fBquntimeout\fR        Solaris DDI
209 \fBqwait\fR     Solaris DDI
210 \fBqwait_sig\fR Solaris DDI
211 \fBqwriter\fR   Solaris DDI
212 \fBRD\fR        DDI/DKI
213 \fBrmvb\fR      DDI/DKI
214 \fBrmvq\fR      DDI/DKI
215 \fBSAMESTR\fR   DDI/DKI
216 \fBstrlog\fR    DDI/DKI
217 \fBstrqget\fR   DDI/DKI
218 \fBstrqset\fR   DDI/DKI
219 \fBtestb\fR     DDI/DKI
220 \fBunbufcall\fR DDI/DKI
221 \fBunfreezestr\fR       DDI/DKI
222 \fBunlinkb\fR   DDI/DKI
223 \fBWR\fR        DDI/DKI
224 .TE

226 .sp
227 .LP
228 The following table summarizes the functions not specific to STREAMS.
229 .sp

231 .sp
232 .TS
233 c c
234 l l .
235 Routine Type
236 _
237 \fBASSERT\fR    DDI/DKI
238 \fBanocancel\fR Solaris DDI
239 \fBaphysio\fR   Solaris DDI
240 \fBatomic_add\fR        DDI/DKI
241 \fBatomic_and\fR        DDI/DKI
242 \fBatomic_bits\fR       DDI/DKI
243 \fBatomic_cas\fR        DDI/DKI
244 \fBatomic_dec\fR        DDI/DKI
245 \fBatomic_inc\fR        DDI/DKI
246 \fBatomic_ops\fR        DDI/DKI
247 \fBatomic_or\fR DDI/DKI
248 \fBatomic_swap\fR       DDI/DKI
249 \fBbcmp\fR      DDI/DKI
250 \fBbcopy\fR     DDI/DKI
251 \fBbioclone\fR  Solaris DDI
252 \fBbiodone\fR   DDI/DKI
253 \fBbiofini\fR   Solaris DDI
254 \fBbioinit\fR   Solaris DDI
255 \fBbiomodified\fR       Solaris DDI
256 \fBbiosize\fR   Solaris DDI
```

```
\fBbioerror\fR  Solaris DDI
\fBbioreset\fR  Solaris DDI
\fBbiowait\fR   DDI/DKI
\fBbp_copyin\fR DDI/DKI
\fBbp_copyout\fR        DDI/DKI
\fBbp_mapin\fR  DDI/DKI
\fBbp_mapout\fR DDI/DKI
\fBbtop\fR      DDI/DKI
\fBbtopr\fR     DDI/DKI
\fBbzero\fR     DDI/DKI
\fBcmn_err\fR   DDI/DKI
\fBcondvar\fR   Solaris DDI
\fBcopyin\fR    DDI/DKI
\fBcopyout\fR   DDI/DKI
\fBcsx_AccessConfigurationRegister\fR   Solaris DDI
\fBcsx_ConvertSize\fR   Solaris DDI
\fBcsx_ConvertSpeed\fR  Solaris DDI
\fBcsx_CS_DDI_Info\fR   Solaris DDI
\fBcsx_DeregisterClient\fR      Solaris DDI
\fBcsx_DupHandle\fR     Solaris DDI
\fBcsx_Error2Text\fR    Solaris DDI
\fBcsx_Event2Text\fR    Solaris DDI
\fBcsx_FreeHandle\fR    Solaris DDI
\fBcsx_Get8\fR  Solaris DDI
\fBcsx_GetFirstClient\fR        Solaris DDI
\fBcsx_GetFirstTuple\fR Solaris DDI
\fBcsx_GetHandleOffset\fR       Solaris DDI
\fBcsx_GetMappedAddr\fR Solaris DDI
\fBcsx_GetStatus\fR     Solaris DDI
\fBcsx_GetTupleData\fR  Solaris DDI
\fBcsx_MakeDeviceNode\fR        Solaris DDI
\fBcsx_MapLogSocket\fR  Solaris DDI
\fBcsx_MapMemPage\fR    Solaris DDI
\fBcsx_ModifyConfiguration\fR   Solaris DDI
\fBcsx_ModifyWindow\fR  Solaris DDI
\fBcsx_Parse_CISTPL_BATTERY\fR  Solaris DDI
\fBcsx_Parse_CISTPL_BYTEORDER\fR        Solaris DDI
\fBcsx_Parse_CISTPL_CFTABLE_ENTRY\fR    Solaris DDI
\fBcsx_Parse_CISTPL_CONFIG\fR   Solaris DDI
\fBcsx_Parse_CISTPL_DATE\fR     Solaris DDI
\fBcsx_Parse_CISTPL_DEVICE\fR   Solaris DDI
\fBcsx_Parse_CISTPL_DEVICEGEO\fR        Solaris DDI
\fBcsx_Parse_CISTPL_DEVICEGEO_A\fR      Solaris DDI
\fBcsx_Parse_CISTPL_FORMAT\fR   Solaris DDI
\fBcsx_Parse_CISTPL_FUNCE\fR    Solaris DDI
\fBcsx_Parse_CISTPL_FUNCID\fR   Solaris DDI
\fBcsx_Parse_CISTPL_GEOMETRY\fR Solaris DDI
\fBcsx_Parse_CISTPL_JEDEC_C\fR  Solaris DDI
\fBcsx_Parse_CISTPL_LINKTARGET\fR       Solaris DDI
\fBcsx_Parse_CISTPL_LONGLINK_A\fR       Solaris DDI
\fBcsx_Parse_CISTPL_LONGLINK_MFC\fR     Solaris DDI
\fBcsx_Parse_CISTPL_MANFID\fR   Solaris DDI
\fBcsx_Parse_CISTPL_ORG\fR      Solaris DDI
\fBcsx_Parse_CISTPL_SPCL\fR     Solaris DDI
\fBcsx_Parse_CISTPL_SWIL\fR     Solaris DDI
\fBcsx_Parse_CISTPL_VERS_1\fR   Solaris DDI
\fBcsx_Parse_CISTPL_VERS_2\fR   Solaris DDI
\fBcsx_ParseTuple\fR    Solaris DDI
\fBcsx_Put8\fR  Solaris DDI
\fBcsx_RegisterClient\fR        Solaris DDI
\fBcsx_ReleaseConfiguration\fR  Solaris DDI
\fBcsx_RepGet8\fR       Solaris DDI
\fBcsx_RepPut8\fR       Solaris DDI
\fBcsx_RequestConfiguration\fR  Solaris DDI
\fBcsx_RequestIO\fR     Solaris DDI
\fBcsx_RequestIRQ\fR    Solaris DDI
```



```
\fBcsx_RequestSocketMask\fR     Solaris DDI
\fBcsx_RequestWindow\fR Solaris DDI
\fBcsx_ResetFunction\fR Solaris DDI
\fBcsx_SetEventMask\fR  Solaris DDI
\fBcsx_SetHandleOffset\fR       Solaris DDI
\fBcsx_ValidateCIS\fR   Solaris DDI
\fBcv_broadcast\fR      Solaris DDI
\fBcv_destroy\fR        Solaris DDI
\fBcv_init\fR   Solaris DDI
\fBcv_signal\fR Solaris DDI
\fBcv_timedwait\fR      Solaris DDI
\fBcv_wait\fR   Solaris DDI
\fBcv_wait_sig\fR       Solaris DDI
\fBddi_add_event_handler\fR     Solaris DDI
\fBddi_add_intr\fR      Solaris DDI
\fBddi_add_softintr\fR  Solaris DDI
\fBddi_binding_name\fR  Solaris DDI
\fBddi_btop\fR  Solaris DDI
\fBddi_btopr\fR Solaris DDI
\fBddi_can_receive_sig\fR       Solaris DDI
\fBddi_check_acc_handle\fR      Solaris DDI
\fBddi_copyin\fR        Solaris DDI
\fBddi_copyout\fR       Solaris DDI
\fBddi_create_minor_node\fR     Solaris DDI
\fBddi_cred\fR  Solaris DDI
\fBddi_dev_is_sid\fR    Solaris DDI
\fBddi_dev_nintrs\fR    Solaris DDI
\fBddi_dev_nregs\fR     Solaris DDI
\fBddi_dev_regsize\fR   Solaris DDI
\fBddi_device_copy\fR   Solaris DDI
\fBddi_device_zero\fR   Solaris DDI
\fBddi_devmap_segmap\fR Solaris DDI
\fBddi_dma_addr_bind_handle\fR  Solaris DDI
\fBddi_dma_alloc_handle\fR      Solaris DDI
\fBddi_dma_buf_bind_handle\fR   Solaris DDI
\fBddi_dma_burstsizes\fR        Solaris DDI
\fBddi_dma_free_handle\fR       Solaris DDI
\fBddi_dma_getwin\fR    Solaris DDI
\fBddi_dma_mem_alloc\fR Solaris DDI
\fBddi_dma_mem_free\fR  Solaris DDI
\fBddi_dma_nextcookie\fR        Solaris DDI
\fBddi_dma_numwin\fR    Solaris DDI
\fBddi_dma_set_sbus64\fR        Solaris DDI
\fBddi_dma_sync\fR      Solaris DDI
\fBddi_dma_unbind_handle\fR     Solaris DDI
\fBddi_dmae\fR  Solaris x86 DDI
\fBddi_dmae_1stparty\fR Solaris x86 DDI
\fBddi_dmae_alloc\fR    Solaris x86 DDI
\fBddi_dmae_disable\fR  Solaris x86 DDI
\fBddi_dmae_enable\fR   Solaris x86 DDI
\fBddi_dmae_getattr\fR  Solaris x86 DDI
\fBddi_dmae_getcnt\fR   Solaris x86 DDI
\fBddi_dmae_getlim\fR   Solaris x86 DDI
\fBddi_dmae_prog\fR     Solaris x86 DDI
\fBddi_dmae_release\fR  Solaris x86 DDI
\fBddi_dmae_stop\fR     Solaris x86 DDI
\fBddi_driver_major\fR  Solaris DDI
\fBddi_driver_name\fR   Solaris DDI
\fBddi_enter_critical\fR        Solaris DDI
\fBddi_exit_critical\fR Solaris DDI
\fBddi_ffs\fR   Solaris DDI
\fBddi_fls\fR   Solaris DDI
\fBddi_fm_acc_err_clear\fR      Solaris DDI
\fBddi_fm_acc_err_get\fR        Solaris DDI
\fBddi_fm_ereport_post\fR       Solaris DDI
\fBddi_fm_handler_register\fR   Solaris DDI
```

```
\fBddi_fm_init\fR       Solaris DDI
\fBddi_fm_service_impact\fR     Solaris DDI
\fBddi_get16\fR Solaris DDI
\fBddi_get32\fR Solaris DDI
\fBddi_get64\fR Solaris DDI
\fBddi_get8\fR  Solaris DDI
\fBddi_get_cred\fR      Solaris DDI
\fBddi_get_devstate\fR  Solaris DDI
\fBddi_get_driver_private\fR    Solaris DDI
\fBddi_get_eventcookie\fR       Solaris DDI
\fBddi_get_iblock_cookie\fR     Solaris DDI
\fBddi_get_iminor\fR    Solaris DDI
\fBddi_get_instance\fR  Solaris DDI
\fBddi_get_kt_did\fR    Solaris DDI
\fBddi_get_lbolt\fR     Solaris DDI
\fBddi_get_name\fR      Solaris DDI
\fBddi_get_parent\fR    Solaris DDI
\fBddi_get_pid\fR       Solaris DDI
\fBddi_get_soft_iblock_cookie\fR        Solaris DDI
\fBddi_get_soft_state\fR        Solaris DDI
\fBddi_getb\fR  Solaris DDI
\fBddi_getl\fR  Solaris DDI
\fBddi_getll\fR Solaris DDI
\fBddi_getlongprop\fR   Solaris DDI
\fBddi_getlongprop_buf\fR       Solaris DDI
\fBddi_getprop\fR       Solaris DDI
\fBddi_getproplen\fR    Solaris DDI
\fBddi_getw\fR  Solaris DDI
\fBddi_intr_add_handler\fR      Solaris DDI
\fBddi_intr_add_softint\fR      Solaris DDI
\fBddi_intr_alloc\fR    Solaris DDI
\fBddi_intr_block_disable\fR    Solaris DDI
\fBddi_intr_block_enable\fR     Solaris DDI
\fBddi_intr_clr_mask\fR Solaris DDI
\fBddi_intr_dup_handler\fR      Solaris DDI
\fBddi_intr_disable\fR  Solaris DDI
\fBddi_intr_enable\fR   Solaris DDI
\fBddi_intr_free\fR     Solaris DDI
\fBddi_intr_get_cap\fR  Solaris DDI
\fBddi_intr_get_hilevel_pri\fR  Solaris DDI
\fBddi_intr_get_navail\fR       Solaris DDI
\fBddi_intr_get_nintrs\fR       Solaris DDI
\fBddi_intr_get_pending\fR      Solaris DDI
\fBddi_intr_get_pri\fR  Solaris DDI
\fBddi_intr_get_softint_pri\fR  Solaris DDI
\fBddi_intr_get_supported_types\fR      Solaris DDI
\fBddi_intr_remove_handler\fR   Solaris DDI
\fBddi_intr_remove_softint\fR   Solaris DDI
\fBddi_intr_set_cap\fR  Solaris DDI
\fBddi_intr_set_mask\fR Solaris DDI
\fBddi_intr_set_pri\fR  Solaris DDI
\fBddi_intr_set_softint_pri\fR  Solaris DDI
\fBddi_intr_trigger_softint\fR  Solaris DDI
\fBddi_io_get16\fR      Solaris DDI
\fBddi_io_get32\fR      Solaris DDI
\fBddi_io_get8\fR       Solaris DDI
\fBddi_io_getb\fR       Solaris DDI
\fBddi_io_getl\fR       Solaris DDI
\fBddi_io_getw\fR       Solaris DDI
\fBddi_io_put16\fR      Solaris DDI
\fBddi_io_put32\fR      Solaris DDI
\fBddi_io_put8\fR       Solaris DDI
\fBddi_io_putb\fR       Solaris DDI
\fBddi_io_putl\fR       Solaris DDI
\fBddi_io_putw\fR       Solaris DDI
\fBddi_io_rep_get16\fR  Solaris DDI
```



```
\fBddi_io_rep_get32\fR  Solaris DDI
\fBddi_io_rep_get8\fR   Solaris DDI
\fBddi_io_rep_getb\fR   Solaris DDI
\fBddi_io_rep_getl\fR   Solaris DDI
\fBddi_io_rep_getw\fR   Solaris DDI
\fBddi_io_rep_put16\fR  Solaris DDI
\fBddi_io_rep_put32\fR  Solaris DDI
\fBddi_io_rep_put8\fR   Solaris DDI
\fBddi_io_rep_putb\fR   Solaris DDI
\fBddi_io_rep_putl\fR   Solaris DDI
\fBddi_io_rep_putw\fR   Solaris DDI
\fBddi_iomin\fR Solaris DDI
\fBddi_log_sysevent\fR  Solaris DDI
\fBddi_map_regs\fR      Solaris DDI
\fBddi_mapdev\fR        Solaris DDI
\fBddi_mapdev_intercept\fR      Solaris DDI
\fBddi_mapdev_nointercept\fR    Solaris DDI
\fBddi_mapdev_set_device_acc_attr\fR    Solaris DDI
\fBddi_mem_get16\fR     Solaris DDI
\fBddi_mem_get32\fR     Solaris DDI
\fBddi_mem_get64\fR     Solaris DDI
\fBddi_mem_get8\fR      Solaris DDI
\fBddi_mem_getb\fR      Solaris DDI
\fBddi_mem_getl\fR      Solaris DDI
\fBddi_mem_getll\fR     Solaris DDI
\fBddi_mem_getw\fR      Solaris DDI
\fBddi_mem_put16\fR     Solaris DDI
\fBddi_mem_put32\fR     Solaris DDI
\fBddi_mem_put64\fR     Solaris DDI
\fBddi_mem_put8\fR      Solaris DDI
\fBddi_mem_putb\fR      Solaris DDI
\fBddi_mem_putl\fR      Solaris DDI
\fBddi_mem_putll\fR     Solaris DDI
\fBddi_mem_putw\fR      Solaris DDI
\fBddi_mem_rep_get16\fR Solaris DDI
\fBddi_mem_rep_get32\fR Solaris DDI
\fBddi_mem_rep_get64\fR Solaris DDI
\fBddi_mem_rep_get8\fR  Solaris DDI
\fBddi_mem_rep_getb\fR  Solaris DDI
\fBddi_mem_rep_getl\fR  Solaris DDI
\fBddi_mem_rep_getll\fR Solaris DDI
\fBddi_mem_rep_getw\fR  Solaris DDI
\fBddi_mem_rep_put16\fR Solaris DDI
\fBddi_mem_rep_put32\fR Solaris DDI
\fBddi_mem_rep_put64\fR Solaris DDI
\fBddi_mem_rep_put8\fR  Solaris DDI
\fBddi_mem_rep_putb\fR  Solaris DDI
\fBddi_mem_rep_putl\fR  Solaris DDI
\fBddi_mem_rep_putll\fR Solaris DDI
\fBddi_mem_rep_putw\fR  Solaris DDI
\fBddi_mmap_get_model\fR        Solaris DDI
\fBddi_model_convert_from\fR    Solaris DDI
\fBddi_modopen\fR       Solaris DDI
\fBddi_no_info\fR       Solaris DDI
\fBddi_node_name\fR     Solaris DDI
\fBddi_peek16\fR        Solaris DDI
\fBddi_peek32\fR        Solaris DDI
\fBddi_peek64\fR        Solaris DDI
\fBddi_peek8\fR Solaris DDI
\fBddi_peekc\fR Solaris DDI
\fBddi_peekd\fR Solaris DDI
\fBddi_peekl\fR Solaris DDI
\fBddi_peeks\fR Solaris DDI
\fBddi_periodic_add\fR  Solaris DDI
\fBddi_periodic_delete\fR       Solaris DDI
\fBddi_poke16\fR        Solaris DDI
```

```
\fBddi_poke32\fR        Solaris DDI
\fBddi_poke64\fR        Solaris DDI
\fBddi_poke8\fR Solaris DDI
\fBddi_pokec\fR Solaris DDI
\fBddi_poked\fR Solaris DDI
\fBddi_pokel\fR Solaris DDI
\fBddi_pokes\fR Solaris DDI
\fBddi_prop_create\fR   Solaris DDI
\fBddi_prop_exists\fR   Solaris DDI
\fBddi_prop_free\fR     Solaris DDI
\fBddi_prop_get_int\fR  Solaris DDI
\fBddi_prop_lookup\fR   Solaris DDI
\fBddi_prop_lookup_byte_array\fR        Solaris DDI
\fBddi_prop_lookup_int_array\fR Solaris DDI
\fBddi_prop_lookup_string\fR    Solaris DDI
\fBddi_prop_lookup_string_array\fR      Solaris DDI
\fBddi_prop_modify\fR   Solaris DDI
\fBddi_prop_op\fR       Solaris DDI
\fBddi_prop_remove\fR   Solaris DDI
\fBddi_prop_remove_all\fR       Solaris DDI
\fBddi_prop_undefine\fR Solaris DDI
\fBddi_prop_update\fR   Solaris DDI
\fBddi_prop_update_byte_array\fR        Solaris DDI
\fBddi_prop_update_int\fR       Solaris DDI
\fBddi_prop_update_int_array\fR Solaris DDI
\fBddi_prop_update_string\fR    Solaris DDI
\fBddi_prop_update_string_array\fR      Solaris DDI
\fBddi_ptob\fR  Solaris DDI
\fBddi_put16\fR Solaris DDI
\fBddi_put32\fR Solaris DDI
\fBddi_put64\fR Solaris DDI
\fBddi_put8\fR  Solaris DDI
\fBddi_putb\fR  Solaris DDI
\fBddi_putl\fR  Solaris DDI
\fBddi_putll\fR Solaris DDI
\fBddi_putw\fR  Solaris DDI
\fBddi_regs_map_free\fR Solaris DDI
\fBddi_regs_map_setup\fR        Solaris DDI
\fBddi_remove_event_handler\fR  Solaris DDI
\fBddi_remove_intr\fR   Solaris DDI
\fBddi_remove_minor_node\fR     Solaris DDI
\fBddi_remove_softintr\fR       Solaris DDI
\fBddi_removing_power\fR        Solaris DDI
\fBddi_rep_get16\fR     Solaris DDI
\fBddi_rep_get32\fR     Solaris DDI
\fBddi_rep_get64\fR     Solaris DDI
\fBddi_rep_get8\fR      Solaris DDI
\fBddi_rep_getb\fR      Solaris DDI
\fBddi_rep_getl\fR      Solaris DDI
\fBddi_rep_getll\fR     Solaris DDI
\fBddi_rep_getw\fR      Solaris DDI
\fBddi_rep_put16\fR     Solaris DDI
\fBddi_rep_put32\fR     Solaris DDI
\fBddi_rep_put64\fR     Solaris DDI
\fBddi_rep_put8\fR      Solaris DDI
\fBddi_rep_putb\fR      Solaris DDI
\fBddi_rep_putl\fR      Solaris DDI
\fBddi_rep_putll\fR     Solaris DDI
\fBddi_rep_putw\fR      Solaris DDI
\fBddi_report_dev\fR    Solaris DDI
\fBddi_root_node\fR     Solaris DDI
\fBddi_segmap\fR        Solaris DDI
\fBddi_segmap_setup\fR  Solaris DDI
\fBddi_set_driver_private\fR    Solaris DDI
\fBddi_slaveonly\fR     Solaris DDI
\fBddi_soft_state\fR    Solaris DDI
```



```
\fBddi_soft_state_fini\fR       Solaris DDI
\fBddi_soft_state_free\fR       Solaris DDI
\fBddi_soft_state_init\fR       Solaris DDI
\fBddi_soft_state_zalloc\fR     Solaris DDI
\fBddi_strlol\fR        Solaris DDI
\fBddi_strloul\fR       Solaris DDI
\fBddi_trigger_softintr\fR      Solaris DDI
\fBddi_umem_alloc\fR    Solaris DDI
\fBddi_umem_free\fR     Solaris DDI
\fBddi_umem_iosetup\fR  Solaris DDI
\fBddi_umem_lock\fR     Solaris DDI
\fBddi_unmap_regs\fR    Solaris DDI
\fBdelay\fR     DDI/DKI
\fBdevmap_default_access\fR     Solaris DDI
\fBdevmap_devmem_setup\fR       Solaris DDI
\fBdevmap_do_ctxmgt\fR  Solaris DDI
\fBdevmap_load\fR       Solaris DDI
\fBdevmap_set_ctx_timeout\fR    Solaris DDI
\fBdevmap_setup\fR      Solaris DDI
\fBdevmap_umem_setup\fR Solaris DDI
\fBdevmap_unload\fR     Solaris DDI
\fBdisksort\fR  Solaris DDI
\fBdlbindack\fR Solaris DDI
\fBdrv_getparm\fR       DDI/DKI
\fBdrv_hztousec\fR      DDI/DKI
\fBdrv_priv\fR  DDI/DKI
\fBdrv_usectohz\fR      DDI/DKI
\fBdrv_usecwait\fR      DDI/DKI
\fBfree_pktiopb\fR      Solaris DDI
\fBfreerbuf\fR  DDI/DKI
\fBget_pktiopb\fR       Solaris DDI
\fBgeterror\fR  DDI/DKI
\fBgethrtime\fR DDI/DKI
\fBgetmajor\fR  DDI/DKI
\fBgetminor\fR  DDI/DKI
\fBgetrbuf\fR   DDI/DKI
\fBgld\fR       Solaris DDI
\fBhat_getkpfnum\fR     DKI only
\fBid32_alloc\fR        Solaris DDI
\fBinb\fR       Solaris x86 DDI
\fBinl\fR       Solaris x86 DDI
\fBinw\fR       Solaris x86 DDI
\fBkiconv\fR    Solaris DDI
\fBkiconv_close\fR      Solaris DDI
\fBkiconv_open\fR       Solaris DDI
\fBkiconvstr\fR Solaris DDI
\fBkmem_alloc\fR        DDI/DKI
\fBkmem_cache_create\fR Solaris DDI
\fBkmem_free\fR DDI/DKI
\fBkmem_zalloc\fR       DDI/DKI
\fBkstat_create\fR      Solaris DDI
\fBkstat_delete\fR      Solaris DDI
\fBkstat_install\fR     Solaris DDI
\fBkstat_named_init\fR  Solaris DDI
\fBkstat_queue\fR       Solaris DDI
\fBkstat_runq_back_to_waitq\fR  Solaris DDI
\fBkstat_runq_enter\fR  Solaris DDI
\fBkstat_runq_exit\fR   Solaris DDI
\fBkstat_waitq_enter\fR Solaris DDI
\fBkstat_waitq_exit\fR  Solaris DDI
\fBkstat_waitq_to_runq\fR       Solaris DDI
\fBldi_add_event_handler\fR     Solaris DDI
\fBldi_aread\fR Solaris DDI
\fBldi_devmap\fR        Solaris DDI
\fBldi_dump\fR  Solaris DDI
\fBldi_ev_finalize\fR   Solaris DDI
```

```
\fBldi_ev_get_cookie\fR Solaris DDI
\fBldi_ev_get_type\fR   Solaris DDI
\fBldi_ev_notify\fR     Solaris DDI
\fBldi_ev_register_callbacks\fR Solaris DDI
\fBldi_ev_remove_callbacks\fR   Solaris DDI
\fBldi_get_dev\fR       Solaris DDI
\fBldi_get_eventcookie\fR       Solaris DDI
\fBldi_get_size\fR      Solaris DDI
\fBldi_ident_from_dev\fR        Solaris DDI
\fBldi_ioctl\fR Solaris DDI
\fBldi_open_by_dev\fR   Solaris DDI
\fBldi_poll\fR  Solaris DDI
\fBldi_prop_exists\fR   Solaris DDI
\fBldi_prop_get_int\fR  Solaris DDI
\fBldi_prop_get_lookup_int_array\fR     Solaris DDI
\fBldi_putmsg\fR        Solaris DDI
\fBldi_read\fR  Solaris DDI
\fBldi_remove_event_handler\fR  Solaris DDI
\fBldi_strategy\fR      Solaris DDI
\fBmakecom_g0\fR        Solaris DDI
\fBmakecom_g0_s\fR      Solaris DDI
\fBmakecom_g1\fR        Solaris DDI
\fBmakecom_g5\fR        Solaris DDI
\fBmakedevice\fR        DDI/DKI
\fBmax\fR       DDI/DKI
\fBmax\fR       DDI/DKI
\fBmembar_ops\fR        Solaris DDI
\fBmemchr\fR    Solaris DDI
\fBminphys\fR   Solaris DDI
\fBmod_info\fR  Solaris DDI
\fBmod_install\fR       Solaris DDI
\fBmod_remove\fR        Solaris DDI
\fBmutex_destroy\fR     Solaris DDI
\fBmutex_enter\fR       Solaris DDI
\fBmutex_exit\fR        Solaris DDI
\fBmutex_init\fR        Solaris DDI
\fBmutex_owned\fR       Solaris DDI
\fBmutex_tryenter\fR    Solaris DDI
\fBnochpoll\fR  Solaris DDI
\fBnodev\fR     DDI/DKI
\fBnulldev\fR   DDI/DKI
\fBnumtos\fR    Solaris DDI
\fBnvlist_add_boolean\fR        Solaris DDI
\fBnvlist_alloc\fR      Solaris DDI
\fBnvlist_lookup_boolean\fR     Solaris DDI
\fBnvlist_lookup_nvpair\fR      Solaris DDI
\fBnvlist_next_nvpair\fR        Solaris DDI
\fBnvlist_remove\fR     Solaris DDI
\fBnvlist_value_byte\fR Solaris DDI
\fBoutb\fR      Solaris x86 DDI
\fBoutl\fR      Solaris x86 DDI
\fBoutw\fR      Solaris x86 DDI
\fBpci_config_get16\fR  Solaris DDI
\fBpci_config_get32\fR  Solaris DDI
\fBpci_config_get64\fR  Solaris DDI
\fBpci_config_get8\fR   Solaris DDI
\fBpci_config_getb\fR   Solaris DDI
\fBpci_config_getl\fR   Solaris DDI
\fBpci_config_getw\fR   Solaris DDI
\fBpci_config_put16\fR  Solaris DDI
\fBpci_config_put32\fR  Solaris DDI
\fBpci_config_put64\fR  Solaris DDI
\fBpci_config_put8\fR   Solaris DDI
\fBpci_config_putb\fR   Solaris DDI
\fBpci_config_putl\fR   Solaris DDI
\fBpci_config_putw\fR   Solaris DDI
```



```
\fBpci_config_setup\fR  Solaris DDI
\fBpci_config_teardown\fR       Solaris DDI
\fBpci_ereport_setup\fR Solaris DDI
\fBpci_report_pmcap\fR  Solaris DDI
\fBpci_save_config_regs\fR      Solaris DDI
\fBphysio\fR    Solaris DDI
\fBpm_busy_component\fR Solaris DDI
\fBpm_power_has_changed\fR      Solaris DDI
\fBpm_raise_power\fR    Solaris DDI
\fBpm_trans_check\fR    Solaris DDI
\fBpollwakeup\fR        DDI/DKI
\fBpci_config_teardown\fR       Solaris DDI
\fBpci_config_teardown\fR       Solaris DDI
\fBpriv_getbyname\fR    Solaris DDI
\fBpriv_policy\fR       Solaris DDI
\fBproc_signal\fR       Solaris DDI
\fBproc_unref\fR        Solaris DDI
\fBptob\fR      DDI/DKI
\fBrepinsb\fR   Solaris x86 DDI
\fBrepinsd\fR   Solaris x86 DDI
\fBrepinsw\fR   Solaris x86 DDI
\fBrepoutsb\fR  Solaris x86 DDI
\fBrepoutsd\fR  Solaris x86 DDI
\fBrepoutsw\fR  Solaris x86 DDI
\fBrmalloc\fR   DDI/DKI
\fBrmalloc_wait\fR      DDI/DKI
\fBrmallocmap\fR        DDI/DKI
\fBrmallocmap_wait\fR   DDI/DKI
\fBrmfree\fR    DDI/DKI
\fBrmfreemap\fR DDI/DKI
\fBrw_destroy\fR        Solaris DDI
\fBrw_downgrade\fR      Solaris DDI
\fBrw_enter\fR  Solaris DDI
\fBrw_exit\fR   Solaris DDI
\fBrw_init\fR   Solaris DDI
\fBrw_read_locked\fR    Solaris DDI
\fBrw_tryenter\fR       Solaris DDI
\fBrw_tryupgrade\fR     Solaris DDI
\fBscsi_abort\fR        Solaris DDI
\fBscsi_alloc_consistent_buf\fR Solaris DDI
\fBscsi_cname\fR        Solaris DDI
\fBscsi_destroy_pkt\fR  Solaris DDI
\fBscsi_dmafree\fR      Solaris DDI
\fBscsi_dmaget\fR       Solaris DDI
\fBscsi_dname\fR        Solaris DDI
\fBscsi_errmsg\fR       Solaris DDI
\fBscsi_ext_sense_fields\fR     Solaris DDI
\fBscsi_find_sense_descr\fR     Solaris DDI
\fBscsi_free_consistent_buf\fR  Solaris DDI
\fBscsi_get_device_type_scsi_options\fR Solaris DDI
\fBscsi_get_device_type_string\fR       Solaris DDI
\fIfBscsi_hba_attach\fR   Solaris DDI
\fBscsi_hba_attach_setup\fR     Solaris DDI
\fBscsi_hba_detach\fR   Solaris DDI
\fBscsi_hba_fini\fR     Solaris DDI
\fBscsi_hba_init\fR     Solaris DDI
\fBscsi_hba_lookup_capstr\fR    Solaris DDI
\fBscsi_hba_pkt_alloc\fR        Solaris DDI
\fBscsi_hba_pkt_free\fR Solaris DDI
\fBscsi_hba_probe\fR    Solaris DDI
\fBscsi_hba_tran_alloc\fR       Solaris DDI
\fBscsi_hba_tran_free\fR        Solaris DDI
\fBscsi_ifgetcap\fR     Solaris DDI
\fBscsi_ifsetcap\fR     Solaris DDI
\fBscsi_init_pkt\fR     Solaris DDI
\fBscsi_log\fR  Solaris DDI
```

```
 784 \fBscsi_mname\fR        Solaris DDI
 785 \fBscsi_pktalloc\fR     Solaris DDI
 786 \fBscsi_pktfree\fR      Solaris DDI
 787 \fBscsi_poll\fR Solaris DDI
 788 \fBscsi_probe\fR        Solaris DDI
 789 \fBscsi_resalloc\fR     Solaris DDI
 790 \fBscsi_reset\fR        Solaris DDI
 791 \fBscsi_reset_notify\fR Solaris DDI
 792 \fBscsi_resfree\fR      Solaris DDI
 793 \fBscsi_rname\fR        Solaris DDI
 794 \fBscsi_sense_key\fR    Solaris DDI
 795 \fBscsi_setup_cdb\fR    Solaris DDI
 796 \fBscsi_slave\fR        Solaris DDI
 797 \fBscsi_sname\fR        Solaris DDI
 798 \fBscsi_sync_pkt\fR     Solaris DDI
 799 \fBscsi_transport\fR    Solaris DDI
 800 \fBscsi_unprobe\fR      Solaris DDI
 801 \fBscsi_unslave\fR      Solaris DDI
 802 \fBscsi_validate_sense\fR       Solaris DDI
 803 \fBscsi_vu_errmsg\fR    Solaris DDI
 804 \fBsema_destroy\fR      Solaris DDI
 805 \fBsema_init\fR Solaris DDI
 806 \fBsema_p\fR    Solaris DDI
 807 \fBsema_p_sig\fR        Solaris DDI
 808 \fBsema_tryp\fR Solaris DDI
 809 \fBsema_v\fR    Solaris DDI
 810 \fBsprintf\fR   Solaris DDI
 811 \fBstoi\fR      Solaris DDI
 812 \fBstrchr\fR    Solaris DDI
 813 \fBstrcmp\fR    Solaris DDI
 814 \fBstrcpy\fR    Solaris DDI
 815 \fBstrlen\fR    Solaris DDI
 816 \fBstrncmp\fR   Solaris DDI
 817 \fBstrncpy\fR   Solaris DDI
 818 \fBSTRUCT_DECL\fR       Solaris DDI
 819 \fBswab\fR      DDI/DKI
 820 \fBtaskq\fR     Solaris DDI
 821 \fBtimeout\fR   DDI/DKI
 822 \fBu8_strcmp\fR Solaris DDI
 823 \fBu8_textprep_str\fR   Solaris DDI
 824 \fBu8_validate\fR       Solaris DDI
 825 \fBuconv_u16tou32\fR    Solaris DDI
 826 \fBuiomove\fR   DDI/DKI
 827 \fBuntimeout\fR DDI/DKI
 828 \fBureadc\fR    DDI/DKI
 829 \fBusb_alloc_request\fR Solaris DDI
 830 \fBusb_client_attach\fR Solaris DDI
 831 \fBusb_clr_feature\fR   Solaris DDI
 832 \fBusb_create_pm_components\fR  Solaris DDI
 833 \fBusb_get_addr\fR      Solaris DDI
 834 \fBusb_get_alt_if\fR    Solaris DDI
 835 \fBusb_get_cfg\fR       Solaris DDI
 836 \fBusb_get_current_frame_number\fR      Solaris DDI
 837 \fBusb_get_dev_data\fR  Solaris DDI
 838 \fBusb_get_max_pkts_per_ioc_request\fR  Solaris DDI
 839 \fBusb_get_status\fR    Solaris DDI
 840 \fBusb_get_string_desc\fR       Solaris DDI
 841 \fBusb_handle_remote_wakeup\fR  Solaris DDI
 842 \fBusb_lookup_ep_data\fR        Solaris DDI
 843 \fBusb_parse_data\fR    Solaris DDI
 844 \fBusb_pipe_bulk_xfer\fR        Solaris DDI
 845 \fBusb_pipe_close\fR    Solaris DDI
 846 \fBusb_pipe_ctrl_xfer\fR        Solaris DDI
 847 \fBusb_pipe_drain_reqs\fR       Solaris DDI
 848 \fBusb_pipe_get_max_bulk_transfer_size\fR       Solaris DDI
 849 \fBusb_pipe_get_state\fR        Solaris DDI
```



```
 850 \fBusb_pipe_intr_xfer\fR        Solaris DDI
 851 \fBusb_pipe_isoc_xfer\fR        Solaris DDI
 852 \fBusb_pipe_open\fR     Solaris DDI
 853 \fBusb_pipe_reset\fR    Solaris DDI
 854 \fBusb_pipe_set_private\fR      Solaris DDI
 855 \fBusb_register_hotplug_cbs\fR  Solaris DDI
 856 \fBusb_reset_device\fR  Solaris DDI
 857 \fBuwritec\fR   DDI/DKI
 858 \fBva_arg\fR    Solaris DDI
 859 \fBva_end\fR    Solaris DDI
 860 \fBva_start\fR  Solaris DDI
 861 \fBvcmn_err\fR  DDI/DKI
 862 \fBvsprintf\fR  Solaris DDI
 863 .TE

 865 .SH SEE ALSO
 866 .sp
 867 .LP
 868 \fBIntro\fR(9E), \fBmutex\fR(9F)
```

```
**********************************************************
   62909 Wed Feb 26 14:23:23 2014
new/usr/src/man/man9f/Makefile
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
   1 #
   2 # This file and its contents are supplied under the terms of the
   3 # Common Development and Distribution License ("CDDL"), version 1.0.
   4 # You may only use this file in accordance with the terms of version
   5 # 1.0 of the CDDL.
   6 #
   7 # A full copy of the text of the CDDL should have accompanied this
   8 # source.  A copy of the CDDL is also available via the Internet at
   9 # http://www.illumos.org/license/CDDL.
  10 #

  12 #
  13 # Copyright 2011, Richard Lowe
  14 # Copyright 2012 Garrett D'Amore <garrett@damore>.
  14 # Copyright 2012 Garrett D'Amore <garrett@damore>.  All rights reserved.
  15 # Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
  16 # Copyright 2014 Pluribus Networks, Inc.
  17 #

  19 include         $(SRC)/Makefile.master

  21 MANSECT=        9f

  23 MANFILES=       ASSERT.9f                               \
  24                 Intro.9f                                \
  25                 OTHERQ.9f                               \
  26                 RD.9f                                   \
  27                 SAMESTR.9f                              \
  28                 STRUCT_DECL.9f                          \
  29                 WR.9f                                   \
  30                 adjmsg.9f                               \
  31                 allocb.9f                               \
  32                 atomic_add.9f                           \
  33                 atomic_and.9f                           \
  34                 atomic_bits.9f                          \
  35                 atomic_cas.9f                           \
  36                 atomic_dec.9f                           \
  37                 atomic_inc.9f                           \
  38                 atomic_ops.9f                           \
  39                 atomic_or.9f                            \
  40                 atomic_swap.9f                          \
  41                 backq.9f                                \
  42                 bcanput.9f                              \
  43                 bcmp.9f                                 \
  44                 bcopy.9f                                \
  45                 bioclone.9f                             \
  46                 biodone.9f                              \
  47                 bioerror.9f                             \
  48                 biofini.9f                              \
  49                 bioinit.9f                              \
  50                 biomodified.9f                          \
  51                 bioreset.9f                             \
  52                 biosize.9f                              \
  53                 biowait.9f                              \
  54                 bp_copyin.9f                            \
  55                 bp_copyout.9f                           \
  56                 bp_mapin.9f                             \
  57                 bp_mapout.9f                            \
  58                 btop.9f                                 \
  59                 btopr.9f                                \
```



```
  60                 bufcall.9f                              \
  61                 bzero.9f                                \
  62                 canput.9f                               \
  63                 clrbuf.9f                               \
  64                 cmn_err.9f                              \
  65                 condvar.9f                              \
  66                 copyb.9f                                \
  67                 copyin.9f                               \
  68                 copymsg.9f                              \
  69                 copyout.9f                              \
  70                 csx_AccessConfigurationRegister.9f      \
  71                 csx_CS_DDI_Info.9f                      \
  72                 csx_ConvertSize.9f                      \
  73                 csx_ConvertSpeed.9f                     \
  74                 csx_DeregisterClient.9f                 \
  75                 csx_DupHandle.9f                        \
  76                 csx_Error2Text.9f                       \
  77                 csx_Event2Text.9f                       \
  78                 csx_FreeHandle.9f                       \
  79                 csx_Get8.9f                             \
  80                 csx_GetFirstClient.9f                   \
  81                 csx_GetFirstTuple.9f                    \
  82                 csx_GetHandleOffset.9f                  \
  83                 csx_GetMappedAddr.9f                    \
  84                 csx_GetStatus.9f                        \
  85                 csx_GetTupleData.9f                     \
  86                 csx_MakeDeviceNode.9f                   \
  87                 csx_MapLogSocket.9f                     \
  88                 csx_MapMemPage.9f                       \
  89                 csx_ModifyConfiguration.9f              \
  90                 csx_ModifyWindow.9f                     \
  91                 csx_ParseTuple.9f                       \
  92                 csx_Parse_CISTPL_BATTERY.9f             \
  93                 csx_Parse_CISTPL_BYTEORDER.9f           \
  94                 csx_Parse_CISTPL_CFTABLE_ENTRY.9f       \
  95                 csx_Parse_CISTPL_CONFIG.9f              \
  96                 csx_Parse_CISTPL_DATE.9f                \
  97                 csx_Parse_CISTPL_DEVICE.9f              \
  98                 csx_Parse_CISTPL_DEVICEGEO.9f           \
  99                 csx_Parse_CISTPL_DEVICEGEO_A.9f         \
 100                 csx_Parse_CISTPL_FORMAT.9f              \
 101                 csx_Parse_CISTPL_FUNCE.9f               \
 102                 csx_Parse_CISTPL_FUNCID.9f              \
 103                 csx_Parse_CISTPL_GEOMETRY.9f            \
 104                 csx_Parse_CISTPL_JEDEC_C.9f             \
 105                 csx_Parse_CISTPL_LINKTARGET.9f          \
 106                 csx_Parse_CISTPL_LONGLINK_A.9f          \
 107                 csx_Parse_CISTPL_LONGLINK_MFC.9f        \
 108                 csx_Parse_CISTPL_MANFID.9f              \
 109                 csx_Parse_CISTPL_ORG.9f                 \
 110                 csx_Parse_CISTPL_SPCL.9f                \
 111                 csx_Parse_CISTPL_SWIL.9f                \
 112                 csx_Parse_CISTPL_VERS_1.9f              \
 113                 csx_Parse_CISTPL_VERS_2.9f              \
 114                 csx_Put8.9f                             \
 115                 csx_RegisterClient.9f                   \
 116                 csx_ReleaseConfiguration.9f             \
 117                 csx_RepGet8.9f                          \
 118                 csx_RepPut8.9f                          \
 119                 csx_RequestConfiguration.9f             \
 120                 csx_RequestIO.9f                        \
 121                 csx_RequestIRQ.9f                       \
 122                 csx_RequestSocketMask.9f                \
 123                 csx_RequestWindow.9f                    \
 124                 csx_ResetFunction.9f                    \
 125                 csx_SetEventMask.9f                     \
```

```
126                csx_SetHandleOffset.9f                \
127                csx_ValidateCIS.9f                    \
128                datamsg.9f                            \
129                ddi_add_event_handler.9f              \
130                ddi_add_intr.9f                       \
131                ddi_add_softintr.9f                   \
132                ddi_binding_name.9f                   \
133                ddi_btop.9f                           \
134                ddi_can_receive_sig.9f                \
135                ddi_cb_register.9f                    \
136                ddi_check_acc_handle.9f               \
137                ddi_copyin.9f                         \
138                ddi_copyout.9f                        \
139                ddi_create_minor_node.9f              \
140                ddi_cred.9f                           \
141                ddi_dev_is_needed.9f                  \
142                ddi_dev_is_sid.9f                     \
143                ddi_dev_nintrs.9f                     \
144                ddi_dev_nregs.9f                      \
145                ddi_dev_regsize.9f                    \
146                ddi_dev_report_fault.9f               \
147                ddi_device_copy.9f                    \
148                ddi_device_zero.9f                    \
149                ddi_devid_compare.9f                  \
150                ddi_dma_addr_bind_handle.9f           \
151                ddi_dma_alloc_handle.9f               \
152                ddi_dma_buf_bind_handle.9f            \
153                ddi_dma_burstsizes.9f                 \
154                ddi_dma_free_handle.9f                \
155                ddi_dma_getwin.9f                     \
156                ddi_dma_mem_alloc.9f                  \
157                ddi_dma_mem_free.9f                   \
158                ddi_dma_nextcookie.9f                 \
159                ddi_dma_numwin.9f                     \
160                ddi_dma_set_sbus64.9f                 \
161                ddi_dma_sync.9f                       \
162                ddi_dma_unbind_handle.9f              \
163                ddi_dmae.9f                           \
164                ddi_driver_major.9f                   \
165                ddi_driver_name.9f                    \
166                ddi_enter_critical.9f                 \
167                ddi_ffs.9f                            \
168                ddi_fm_acc_err_clear.9f               \
169                ddi_fm_acc_err_get.9f                 \
170                ddi_fm_ereport_post.9f                \
171                ddi_fm_handler_register.9f            \
172                ddi_fm_init.9f                        \
173                ddi_fm_service_impact.9f              \
174                ddi_get8.9f                           \
175                ddi_get_cred.9f                       \
176                ddi_get_devstate.9f                   \
177                ddi_get_driver_private.9f             \
178                ddi_get_eventcookie.9f                \
179                ddi_get_instance.9f                   \
180                ddi_get_kt_did.9f                     \
181                ddi_get_lbolt.9f                      \
182                ddi_get_parent.9f                     \
183                ddi_get_pid.9f                        \
184                ddi_get_time.9f                       \
185                ddi_getiminor.9f                      \
186                ddi_in_panic.9f                       \
187                ddi_intr_add_handler.9f               \
188                ddi_intr_add_softint.9f               \
189                ddi_intr_alloc.9f                     \
190                ddi_intr_dup_handler.9f               \
191                ddi_intr_enable.9f                    \
```



```
192                ddi_intr_get_cap.9f                   \
193                ddi_intr_get_hilevel_pri.9f           \
194                ddi_intr_get_nintrs.9f                \
195                ddi_intr_get_pending.9f               \
196                ddi_intr_get_pri.9f                   \
197                ddi_intr_get_supported_types.9f       \
198                ddi_intr_hilevel.9f                   \
199                ddi_intr_set_mask.9f                  \
200                ddi_intr_set_nreq.9f                  \
201                ddi_io_get8.9f                        \
202                ddi_io_put8.9f                        \
203                ddi_io_rep_get8.9f                    \
204                ddi_io_rep_put8.9f                    \
205                ddi_iomin.9f                          \
206                ddi_log_sysevent.9f                   \
207                ddi_map_regs.9f                       \
208                ddi_mem_get8.9f                       \
209                ddi_mem_put8.9f                       \
210                ddi_mem_rep_get8.9f                   \
211                ddi_mem_rep_put8.9f                   \
212                ddi_mmap_get_model.9f                 \
213                ddi_model_convert_from.9f             \
214                ddi_modopen.9f                        \
215                ddi_no_info.9f                        \
216                ddi_node_name.9f                      \
217                ddi_peek.9f                           \
218                ddi_periodic_add.9f                   \
219                ddi_periodic_delete.9f                \
220                ddi_poke.9f                           \
221                ddi_prop_create.9f                    \
222                ddi_prop_exists.9f                    \
223                ddi_prop_get_int.9f                   \
224                ddi_prop_lookup.9f                    \
225                ddi_prop_op.9f                        \
226                ddi_prop_update.9f                    \
227                ddi_put8.9f                           \
228                ddi_regs_map_free.9f                  \
229                ddi_regs_map_setup.9f                 \
230                ddi_remove_event_handler.9f           \
231                ddi_remove_minor_node.9f              \
232                ddi_removing_power.9f                 \
233                ddi_rep_get8.9f                       \
234                ddi_rep_put8.9f                       \
235                ddi_report_dev.9f                     \
236                ddi_root_node.9f                      \
237                ddi_segmap.9f                         \
238                ddi_slaveonly.9f                      \
239                ddi_soft_state.9f                     \
240                ddi_strtol.9f                         \
241                ddi_strtoll.9f                        \
242                ddi_strtoul.9f                        \
243                ddi_umem_alloc.9f                     \
244                ddi_umem_iosetup.9f                   \
245                ddi_umem_lock.9f                      \
246                delay.9f                              \
247                devmap_default_access.9f              \
248                devmap_devmem_setup.9f                \
249                devmap_do_ctxmgt.9f                   \
250                devmap_set_ctx_timeout.9f             \
251                devmap_setup.9f                       \
252                devmap_unload.9f                      \
253                disksort.9f                           \
254                dlbindack.9f                          \
255                drv_getparm.9f                        \
256                drv_hztousec.9f                       \
257                drv_priv.9f                           \
```

```
258                drv_usectohz.9f                 \
259                drv_usecwait.9f                 \
260                dupb.9f                         \
261                dupmsg.9f                       \
262                enableok.9f                     \
263                esballoc.9f                     \
264                esbbcall.9f                     \
265                flushband.9f                    \
266                flushq.9f                       \
267                freeb.9f                        \
268                freemsg.9f                      \
269                freerbuf.9f                     \
270                freezestr.9f                    \
271                get_pktiopb.9f                  \
272                geterror.9f                     \
273                gethrtime.9f                    \
274                getmajor.9f                     \
275                getminor.9f                     \
276                getq.9f                         \
277                getrbuf.9f                      \
278                gld.9f                          \
279                hook_alloc.9f                   \
280                hook_free.9f                    \
281                id32_alloc.9f                   \
282                inb.9f                          \
283                insq.9f                         \
284                kiconv.9f                       \
285                kiconv_close.9f                 \
286                kiconv_open.9f                  \
287                kiconvstr.9f                    \
288                kmem_alloc.9f                   \
289                kmem_cache_create.9f            \
290                kstat_create.9f                 \
291                kstat_delete.9f                 \
292                kstat_install.9f                \
293                kstat_named_init.9f             \
294                kstat_queue.9f                  \
295                ldi_add_event_handler.9f        \
296                ldi_aread.9f                    \
297                ldi_devmap.9f                   \
298                ldi_dump.9f                     \
299                ldi_ev_finalize.9f              \
300                ldi_ev_get_cookie.9f            \
301                ldi_ev_get_type.9f              \
302                ldi_ev_notify.9f                \
303                ldi_ev_register_callbacks.9f    \
304                ldi_ev_remove_callbacks.9f      \
305                ldi_get_dev.9f                  \
306                ldi_get_eventcookie.9f          \
307                ldi_get_size.9f                 \
308                ldi_ident_from_dev.9f           \
309                ldi_ioctl.9f                    \
310                ldi_open_by_dev.9f              \
311                ldi_poll.9f                     \
312                ldi_prop_exists.9f              \
313                ldi_prop_get_int.9f             \
314                ldi_prop_lookup_int_array.9f    \
315                ldi_putmsg.9f                   \
316                ldi_read.9f                     \
317                ldi_remove_event_handler.9f     \
318                ldi_strategy.9f                 \
319                linkb.9f                        \
320                list_create.9f                  \
321                makecom.9f                      \
322                makedevice.9f                   \
323                max.9f                          \
```



```
324                mcopyin.9f                      \
325                mcopymsg.9f                     \
326                mcopyout.9f                     \
327                membar_ops.9f                   \
328                memchr.9f                       \
329                merror.9f                       \
330                mexchange.9f                    \
331                min.9f                          \
332                mioc2ack.9f                     \
333                miocack.9f                      \
334                miocnak.9f                      \
335                miocpullup.9f                   \
336                mkiocb.9f                       \
337                mod_install.9f                  \
338                msgdsize.9f                     \
339                msgpullup.9f                    \
340                msgsize.9f                      \
341                mt-streams.9f                   \
342                mutex.9f                        \
343                net_event_notify_register.9f    \
344                net_getifname.9f                \
345                net_getlifaddr.9f               \
346                net_getmtu.9f                   \
347                net_getnetid.9f                 \
348                net_getpmtuenabled.9f           \
349                net_hook_register.9f            \
350                net_hook_unregister.9f          \
351                net_inject.9f                   \
352                net_inject_alloc.9f             \
353                net_inject_free.9f              \
354                net_instance_alloc.9f           \
355                net_instance_free.9f            \
356                net_instance_notify_register.9f \
357                net_instance_register.9f        \
358                net_instance_unregister.9f      \
359                net_ispartialchecksum.9f        \
360                net_isvalidchecksum.9f          \
361                net_kstat_create.9f             \
362                net_kstat_delete.9f             \
363                net_lifgetnext.9f               \
364                net_netidtozonid.9f             \
365                net_phygetnext.9f               \
366                net_phylookup.9f                \
367                net_protocol_lookup.9f          \
368                net_protocol_notify_register.9f \
369                net_protocol_release.9f         \
370                net_protocol_walk.9f            \
371                net_routeto.9f                  \
372                net_zoneidtonetid.9f            \
373                netinfo.9f                      \
374                nochpoll.9f                     \
375                nodev.9f                        \
376                noenable.9f                     \
377                nulldev.9f                      \
378                nvlist_add_boolean.9f           \
379                nvlist_alloc.9f                 \
380                nvlist_lookup_boolean.9f        \
381                nvlist_lookup_nvpair.9f         \
382                nvlist_next_nvpair.9f           \
383                nvlist_remove.9f                \
384                nvpair_value_byte.9f            \
385                outb.9f                         \
386                pci_config_get8.9f              \
387                pci_config_setup.9f             \
388                pci_ereport_setup.9f            \
389                pci_report_pmcap.9f             \
```

```
390                 pci_save_config_regs.9f                 \
391                 physio.9f                               \
392                 pm_busy_component.9f                    \
393                 pm_power_has_changed.9f                 \
394                 pm_raise_power.9f                       \
395                 pm_trans_check.9f                       \
396                 pollwakeup.9f                           \
397                 priv_getbyname.9f                       \
398                 priv_policy.9f                          \
399                 proc_signal.9f                          \
400                 ptob.9f                                 \
401                 pullupmsg.9f                            \
402                 put.9f                                  \
403                 putbq.9f                                \
404                 putctl.9f                               \
405                 putctl1.9f                              \
406                 putnext.9f                              \
407                 putnextctl.9f                           \
408                 putnextctl1.9f                          \
409                 putq.9f                                 \
410                 qassociate.9f                           \
411                 qbufcall.9f                             \
412                 qenable.9f                              \
413                 qprocson.9f                             \
414                 qreply.9f                               \
415                 qsize.9f                                \
416                 qtimeout.9f                             \
417                 qunbufcall.9f                           \
418                 quntimeout.9f                           \
419                 qwait.9f                                \
420                 qwriter.9f                              \
421                 rmalloc.9f                              \
422                 rmalloc_wait.9f                         \
423                 rmallocmap.9f                           \
424                 rmfree.9f                               \
425                 rmvb.9f                                 \
426                 rmvq.9f                                 \
427                 rwlock.9f                               \
428                 scsi_abort.9f                           \
429                 scsi_alloc_consistent_buf.9f            \
430                 scsi_cname.9f                           \
431                 scsi_destroy_pkt.9f                     \
432                 scsi_dmaget.9f                          \
433                 scsi_errmsg.9f                          \
434                 scsi_ext_sense_fields.9f                \
435                 scsi_find_sense_descr.9f                \
436                 scsi_free_consistent_buf.9f             \
437                 scsi_get_device_type_scsi_options.9f    \
438                 scsi_get_device_type_string.9f          \
439                 scsi_hba_attach_setup.9f                \
440                 scsi_hba_init.9f                        \
441                 scsi_hba_lookup_capstr.9f               \
442                 scsi_hba_pkt_alloc.9f                   \
443                 scsi_hba_pkt_comp.9f                    \
444                 scsi_hba_probe.9f                       \
445                 scsi_hba_tran_alloc.9f                  \
446                 scsi_ifgetcap.9f                        \
447                 scsi_init_pkt.9f                        \
448                 scsi_log.9f                             \
449                 scsi_pktalloc.9f                        \
450                 scsi_poll.9f                            \
451                 scsi_probe.9f                           \
452                 scsi_reset.9f                           \
453                 scsi_reset_notify.9f                    \
454                 scsi_sense_key.9f                       \
455                 scsi_setup_cdb.9f                       \
```



```
456                 scsi_slave.9f                           \
457                 scsi_sync_pkt.9f                        \
458                 scsi_transport.9f                       \
459                 scsi_unprobe.9f                         \
460                 scsi_validate_sense.9f                  \
461                 scsi_vu_errmsg.9f                       \
462                 semaphore.9f                            \
463                 stoi.9f                                 \
464                 string.9f                               \
465                 strlog.9f                               \
466                 strqget.9f                              \
467                 strqset.9f                              \
468                 swab.9f                                 \
469                 taskq.9f                                \
470                 testb.9f                                \
471                 timeout.9f                              \
472                 u8_strcmp.9f                            \
473                 u8_textprep_str.9f                      \
474                 u8_validate.9f                          \
475                 uconv_u16tou32.9f                       \
476                 uiomove.9f                              \
477                 unbufcall.9f                            \
478                 unlinkb.9f                              \
479                 untimeout.9f                            \
480                 ureadc.9f                               \
481                 usb_alloc_request.9f                    \
482                 usb_client_attach.9f                    \
483                 usb_clr_feature.9f                      \
484                 usb_create_pm_components.9f             \
485                 usb_get_addr.9f                         \
486                 usb_get_alt_if.9f                       \
487                 usb_get_cfg.9f                          \
488                 usb_get_current_frame_number.9f         \
489                 usb_get_dev_data.9f                     \
490                 usb_get_max_pkts_per_isoc_request.9f    \
491                 usb_get_status.9f                       \
492                 usb_get_string_descr.9f                 \
493                 usb_handle_remote_wakeup.9f             \
494                 usb_lookup_ep_data.9f                   \
495                 usb_parse_data.9f                       \
496                 usb_pipe_bulk_xfer.9f                   \
497                 usb_pipe_close.9f                       \
498                 usb_pipe_ctrl_xfer.9f                   \
499                 usb_pipe_drain_reqs.9f                  \
500                 usb_pipe_get_max_bulk_transfer_size.9f  \
501                 usb_pipe_get_state.9f                   \
502                 usb_pipe_intr_xfer.9f                   \
503                 usb_pipe_isoc_xfer.9f                   \
504                 usb_pipe_open.9f                        \
505                 usb_pipe_reset.9f                       \
506                 usb_pipe_set_private.9f                 \
507                 usb_register_hotplug_cbs.9f             \
508                 usb_reset_device.9f                     \
509                 uwritec.9f                              \
510                 va_arg.9f                               \
511                 vsprintf.9f

513 MANLINKS=       SIZEOF_PTR.9f                           \
514                 SIZEOF_STRUCT.9f                        \
515                 STRUCT_BUF.9f                           \
516                 STRUCT_FADDR.9f                         \
517                 STRUCT_FGET.9f                          \
518                 STRUCT_FGETP.9f                         \
519                 STRUCT_FSET.9f                          \
520                 STRUCT_FSETP.9f                         \
521                 STRUCT_HANDLE.9f                        \
```

```
522                STRUCT_INIT.9f                  \
523                STRUCT_SET_HANDLE.9f            \
524                STRUCT_SIZE.9f                  \
525                assert.9f                       \
526                atomic_add_16.9f                \
527                atomic_add_16_nv.9f             \
528                atomic_add_32.9f                \
529                atomic_add_32_nv.9f             \
530                atomic_add_64.9f                \
531                atomic_add_64_nv.9f             \
532                atomic_add_8.9f                 \
533                atomic_add_8_nv.9f              \
534                atomic_add_char.9f              \
535                atomic_add_char_nv.9f           \
536                atomic_add_int.9f               \
537                atomic_add_int_nv.9f            \
538                atomic_add_long.9f              \
539                atomic_add_long_nv.9f           \
540                atomic_add_ptr.9f               \
541                atomic_add_ptr_nv.9f            \
542                atomic_add_short.9f             \
543                atomic_add_short_nv.9f          \
544                atomic_and_16.9f                \
545                atomic_and_16_nv.9f             \
546                atomic_and_32.9f                \
547                atomic_and_32_nv.9f             \
548                atomic_and_64.9f                \
549                atomic_and_64_nv.9f             \
550                atomic_and_8.9f                 \
551                atomic_and_8_nv.9f              \
552                atomic_and_uchar.9f             \
553                atomic_and_uchar_nv.9f          \
554                atomic_and_uint.9f              \
555                atomic_and_uint_nv.9f           \
556                atomic_and_ulong.9f             \
557                atomic_and_ulong_nv.9f          \
558                atomic_and_ushort.9f            \
559                atomic_and_ushort_nv.9f         \
560                atomic_cas_16.9f                \
561                atomic_cas_32.9f                \
562                atomic_cas_64.9f                \
563                atomic_cas_8.9f                 \
564                atomic_cas_ptr.9f               \
565                atomic_cas_uchar.9f             \
566                atomic_cas_uint.9f              \
567                atomic_cas_ulong.9f             \
568                atomic_cas_ushort.9f            \
569                atomic_clear_long_excl.9f       \
570                atomic_dec_16.9f                \
571                atomic_dec_16_nv.9f             \
572                atomic_dec_32.9f                \
573                atomic_dec_32_nv.9f             \
574                atomic_dec_64.9f                \
575                atomic_dec_64_nv.9f             \
576                atomic_dec_8.9f                 \
577                atomic_dec_8_nv.9f              \
578                atomic_dec_ptr.9f               \
579                atomic_dec_ptr_nv.9f            \
580                atomic_dec_uchar.9f             \
581                atomic_dec_uchar_nv.9f          \
582                atomic_dec_uint.9f              \
583                atomic_dec_uint_nv.9f           \
584                atomic_dec_ulong.9f             \
585                atomic_dec_ulong_nv.9f          \
586                atomic_dec_ushort.9f            \
587                atomic_dec_ushort_nv.9f         \
```



```
588                atomic_inc_16.9f                \
589                atomic_inc_16_nv.9f             \
590                atomic_inc_32.9f                \
591                atomic_inc_32_nv.9f             \
592                atomic_inc_64.9f                \
593                atomic_inc_64_nv.9f             \
594                atomic_inc_8.9f                 \
595                atomic_inc_8_nv.9f              \
596                atomic_inc_ptr.9f               \
597                atomic_inc_ptr_nv.9f            \
598                atomic_inc_uchar.9f             \
599                atomic_inc_uchar_nv.9f          \
600                atomic_inc_uint.9f              \
601                atomic_inc_uint_nv.9f           \
602                atomic_inc_ulong.9f             \
603                atomic_inc_ulong_nv.9f          \
604                atomic_inc_ushort.9f            \
605                atomic_inc_ushort_nv.9f         \
606                atomic_or_16.9f                 \
607                atomic_or_16_nv.9f              \
608                atomic_or_32.9f                 \
609                atomic_or_32_nv.9f              \
610                atomic_or_64.9f                 \
611                atomic_or_64_nv.9f              \
612                atomic_or_8.9f                  \
613                atomic_or_8_nv.9f               \
614                atomic_or_uchar.9f              \
615                atomic_or_uchar_nv.9f           \
616                atomic_or_uint.9f               \
617                atomic_or_uint_nv.9f            \
618                atomic_or_ulong.9f              \
619                atomic_or_ulong_nv.9f           \
620                atomic_or_ushort.9f             \
621                atomic_or_ushort_nv.9f          \
622                atomic_set_long_excl.9f         \
623                atomic_swap_16.9f               \
624                atomic_swap_32.9f               \
625                atomic_swap_64.9f               \
626                atomic_swap_8.9f                \
627                atomic_swap_ptr.9f              \
628                atomic_swap_uchar.9f            \
629                atomic_swap_uint.9f             \
630                atomic_swap_ulong.9f            \
631                atomic_swap_ushort.9f           \
632                crgetgid.9f                     \
633                crgetgroups.9f                  \
634                crgetngroups.9f                 \
635                crgetrgid.9f                    \
636                crgetruid.9f                    \
637                crgetsgid.9f                    \
638                crgetsuid.9f                    \
639                crgetuid.9f                     \
640                crgetzoneid.9f                  \
641                csx_Get16.9f                    \
642                csx_Get32.9f                    \
643                csx_Get64.9f                    \
644                csx_GetEventMask.9f             \
645                csx_GetNextClient.9f            \
646                csx_GetNextTuple.9f             \
647                csx_Parse_CISTPL_DEVICE_A.9f    \
648                csx_Parse_CISTPL_DEVICE_OA.9f   \
649                csx_Parse_CISTPL_DEVICE_OC.9f   \
650                csx_Parse_CISTPL_JEDEC_A.9f     \
651                csx_Parse_CISTPL_LONGLINK_C.9f  \
652                csx_Put16.9f                    \
653                csx_Put32.9f                    \
```

```
654                     csx_Put64.9f                    \
655                     csx_ReleaseIO.9f                \
656                     csx_ReleaseIRQ.9f               \
657                     csx_ReleaseSocketMask.9f        \
658                     csx_ReleaseWindow.9f            \
659                     csx_RemoveDeviceNode.9f         \
660                     csx_RepGet16.9f                 \
661                     csx_RepGet32.9f                 \
662                     csx_RepGet64.9f                 \
663                     csx_RepPut16.9f                 \
664                     csx_RepPut32.9f                 \
665                     csx_RepPut64.9f                 \
666                     cv_broadcast.9f                 \
667                     cv_destroy.9f                   \
668                     cv_init.9f                      \
669                     cv_reltimedwait.9f              \
670                     cv_reltimedwait_sig.9f          \
671                     cv_signal.9f                    \
672                     cv_timedwait.9f                 \
673                     cv_timedwait_sig.9f             \
674                     cv_wait.9f                      \
675                     cv_wait_sig.9f                  \
676                     ddi_btopr.9f                    \
677                     ddi_cb_unregister.9f            \
678                     ddi_check_dma_handle.9f         \
679                     ddi_devid_free.9f               \
680                     ddi_devid_get.9f                \
681                     ddi_devid_init.9f               \
682                     ddi_devid_register.9f           \
683                     ddi_devid_sizeof.9f             \
684                     ddi_devid_str_decode.9f         \
685                     ddi_devid_str_encode.9f         \
686                     ddi_devid_str_free.9f           \
687                     ddi_devid_unregister.9f         \
688                     ddi_devid_valid.9f              \
689                     ddi_devmap_segmap.9f            \
690                     ddi_dmae_1stparty.9f            \
691                     ddi_dmae_alloc.9f               \
692                     ddi_dmae_disable.9f             \
693                     ddi_dmae_enable.9f              \
694                     ddi_dmae_getattr.9f             \
695                     ddi_dmae_getcnt.9f              \
696                     ddi_dmae_getlim.9f              \
697                     ddi_dmae_prog.9f                \
698                     ddi_dmae_release.9f             \
699                     ddi_dmae_stop.9f                \
700                     ddi_exit_critical.9f            \
701                     ddi_fls.9f                      \
702                     ddi_fm_capable.9f               \
703                     ddi_fm_dma_err_clear.9f         \
704                     ddi_fm_dma_err_get.9f           \
705                     ddi_fm_fini.9f                  \
706                     ddi_fm_handler_unregister.9f    \
707                     ddi_get16.9f                    \
708                     ddi_get32.9f                    \
709                     ddi_get64.9f                    \
710                     ddi_get_iblock_cookie.9f        \
711                     ddi_get_lbolt64.9f              \
712                     ddi_get_name.9f                 \
713                     ddi_get_soft_iblock_cookie.9f   \
714                     ddi_get_soft_state.9f           \
715                     ddi_getb.9f                     \
716                     ddi_getl.9f                     \
717                     ddi_getll.9f                    \
718                     ddi_getlongprop.9f              \
719                     ddi_getlongprop_buf.9f          \
```



```
720                     ddi_getprop.9f                  \
721                     ddi_getproplen.9f               \
722                     ddi_getw.9f                     \
723                     ddi_intr_block_disable.9f       \
724                     ddi_intr_block_enable.9f        \
725                     ddi_intr_clr_mask.9f            \
726                     ddi_intr_disable.9f             \
727                     ddi_intr_free.9f                \
728                     ddi_intr_get_navail.9f          \
729                     ddi_intr_get_softint_pri.9f     \
730                     ddi_intr_remove_handler.9f      \
731                     ddi_intr_remove_softint.9f      \
732                     ddi_intr_set_cap.9f             \
733                     ddi_intr_set_pri.9f             \
734                     ddi_intr_set_softint_pri.9f     \
735                     ddi_intr_trigger_softint.9f     \
736                     ddi_io_get16.9f                 \
737                     ddi_io_get32.9f                 \
738                     ddi_io_getb.9f                  \
739                     ddi_io_getl.9f                  \
740                     ddi_io_getw.9f                  \
741                     ddi_io_put16.9f                 \
742                     ddi_io_put32.9f                 \
743                     ddi_io_putb.9f                  \
744                     ddi_io_putl.9f                  \
745                     ddi_io_putw.9f                  \
746                     ddi_io_rep_get16.9f             \
747                     ddi_io_rep_get32.9f             \
748                     ddi_io_rep_getb.9f              \
749                     ddi_io_rep_getl.9f              \
750                     ddi_io_rep_getw.9f              \
751                     ddi_io_rep_put16.9f             \
752                     ddi_io_rep_put32.9f             \
753                     ddi_io_rep_putb.9f              \
754                     ddi_io_rep_putl.9f              \
755                     ddi_io_rep_putw.9f              \
756                     ddi_mem_get16.9f                \
757                     ddi_mem_get32.9f                \
758                     ddi_mem_get64.9f                \
759                     ddi_mem_getb.9f                 \
760                     ddi_mem_getl.9f                 \
761                     ddi_mem_getll.9f                \
762                     ddi_mem_getw.9f                 \
763                     ddi_mem_put16.9f                \
764                     ddi_mem_put32.9f                \
765                     ddi_mem_put64.9f                \
766                     ddi_mem_putb.9f                 \
767                     ddi_mem_putl.9f                 \
768                     ddi_mem_putll.9f                \
769                     ddi_mem_putw.9f                 \
770                     ddi_mem_rep_get16.9f            \
771                     ddi_mem_rep_get32.9f            \
772                     ddi_mem_rep_get64.9f            \
773                     ddi_mem_rep_getb.9f             \
774                     ddi_mem_rep_getl.9f             \
775                     ddi_mem_rep_getll.9f            \
776                     ddi_mem_rep_getw.9f             \
777                     ddi_mem_rep_put16.9f            \
778                     ddi_mem_rep_put32.9f            \
779                     ddi_mem_rep_put64.9f            \
780                     ddi_mem_rep_putb.9f             \
781                     ddi_mem_rep_putl.9f             \
782                     ddi_mem_rep_putll.9f            \
783                     ddi_mem_rep_putw.9f             \
784                     ddi_modclose.9f                 \
785                     ddi_modsym.9f                   \
```

```
786                 ddi_peek16.9f                       \
787                 ddi_peek32.9f                       \
788                 ddi_peek64.9f                       \
789                 ddi_peek8.9f                        \
790                 ddi_peekc.9f                        \
791                 ddi_peekd.9f                        \
792                 ddi_peekl.9f                        \
793                 ddi_peeks.9f                        \
794                 ddi_poke16.9f                       \
795                 ddi_poke32.9f                       \
796                 ddi_poke64.9f                       \
797                 ddi_poke8.9f                        \
798                 ddi_pokec.9f                        \
799                 ddi_poked.9f                        \
800                 ddi_pokel.9f                        \
801                 ddi_pokes.9f                        \
802                 ddi_prop_free.9f                    \
803                 ddi_prop_get_int64.9f               \
804                 ddi_prop_lookup_byte_array.9f       \
805                 ddi_prop_lookup_int64_array.9f      \
806                 ddi_prop_lookup_int_array.9f        \
807                 ddi_prop_lookup_string.9f           \
808                 ddi_prop_lookup_string_array.9f     \
809                 ddi_prop_modify.9f                  \
810                 ddi_prop_remove.9f                  \
811                 ddi_prop_remove_all.9f              \
812                 ddi_prop_undefine.9f                \
813                 ddi_prop_update_byte_array.9f       \
814                 ddi_prop_update_int.9f              \
815                 ddi_prop_update_int64.9f            \
816                 ddi_prop_update_int64_array.9f      \
817                 ddi_prop_update_int_array.9f        \
818                 ddi_prop_update_string.9f           \
819                 ddi_prop_update_string_array.9f     \
820                 ddi_ptob.9f                         \
821                 ddi_put16.9f                        \
822                 ddi_put32.9f                        \
823                 ddi_put64.9f                        \
824                 ddi_putb.9f                         \
825                 ddi_putl.9f                         \
826                 ddi_putll.9f                        \
827                 ddi_putw.9f                         \
828                 ddi_remove_intr.9f                  \
829                 ddi_remove_softintr.9f              \
830                 ddi_rep_get16.9f                    \
831                 ddi_rep_get32.9f                    \
832                 ddi_rep_get64.9f                    \
833                 ddi_rep_getb.9f                     \
834                 ddi_rep_getl.9f                     \
835                 ddi_rep_getll.9f                    \
836                 ddi_rep_getw.9f                     \
837                 ddi_rep_put16.9f                    \
838                 ddi_rep_put32.9f                    \
839                 ddi_rep_put64.9f                    \
840                 ddi_rep_putb.9f                     \
841                 ddi_rep_putl.9f                     \
842                 ddi_rep_putll.9f                    \
843                 ddi_rep_putw.9f                     \
844                 ddi_segmap_setup.9f                 \
845                 ddi_set_driver_private.9f           \
846                 ddi_soft_state_fini.9f              \
847                 ddi_soft_state_free.9f              \
848                 ddi_soft_state_init.9f              \
849                 ddi_soft_state_zalloc.9f            \
850                 ddi_strdup.9f                       \
851                 ddi_strtoull.9f                     \
```



```
852                 ddi_taskq_create.9f                 \
853                 ddi_taskq_destroy.9f                \
854                 ddi_taskq_dispatch.9f               \
855                 ddi_taskq_resume.9f                 \
856                 ddi_taskq_suspend.9f                \
857                 ddi_taskq_wait.9f                   \
858                 ddi_trigger_softintr.9f             \
859                 ddi_umem_free.9f                    \
860                 ddi_umem_unlock.9f                  \
861                 ddi_unmap_regs.9f                   \
862                 desballoc.9f                        \
863                 devmap_load.9f                      \
864                 devmap_umem_setup.9f                \
865                 dlerrorack.9f                       \
866                 dlokack.9f                          \
867                 dlphysaddrack.9f                    \
868                 dluderrorind.9f                     \
869                 free_pktiopb.9f                     \
870                 gld_intr.9f                         \
871                 gld_mac_alloc.9f                    \
872                 gld_mac_free.9f                     \
873                 gld_recv.9f                         \
874                 gld_register.9f                     \
875                 gld_sched.9f                        \
876                 gld_unregister.9f                   \
877                 id32_free.9f                        \
878                 id32_lookup.9f                      \
879                 inl.9f                              \
880                 intro.9f                            \
881                 inw.9f                              \
882                 kmem_cache_alloc.9f                 \
883                 kmem_cache_destroy.9f               \
884                 kmem_cache_free.9f                  \
885                 kmem_cache_set_move.9f              \
886                 kmem_free.9f                        \
887                 kmem_zalloc.9f                      \
888                 kstat_named_setstr.9f               \
889                 kstat_runq_back_to_waitq.9f         \
890                 kstat_runq_enter.9f                 \
891                 kstat_runq_exit.9f                  \
892                 kstat_waitq_enter.9f                \
893                 kstat_waitq_exit.9f                 \
894                 kstat_waitq_to_runq.9f              \
895                 ldi_awrite.9f                       \
896                 ldi_close.9f                        \
897                 ldi_get_devid.9f                    \
898                 ldi_get_minor_name.9f               \
899                 ldi_get_otyp.9f                     \
900                 ldi_getmsg.9f                       \
901                 ldi_ident_from_dip.9f               \
902                 ldi_ident_from_stream.9f            \
903                 ldi_ident_release.9f                \
904                 ldi_open_by_devid.9f                \
905                 ldi_open_by_name.9f                 \
906                 ldi_prop_get_int64.9f               \
907                 ldi_prop_lookup_byte_array.9f       \
908                 ldi_prop_lookup_int64_array.9f      \
909                 ldi_prop_lookup_string.9f           \
910                 ldi_prop_lookup_string_array.9f     \
911                 ldi_write.9f                        \
912                 list_destroy.9f                     \
913                 list_head.9f                        \
914                 list_insert_after.9f                \
915                 list_insert_before.9f               \
916                 list_insert_head.9f                 \
917                 list_insert_tail.9f                 \
```

```
                list_is_empty.9f                        \
                list_link_active.9f                     \
                list_link_init.9f                       \
                list_link_replace.9f                    \
                list_move_tail.9f                       \
                list_next.9f                            \
                list_prev.9f                            \
                list_remove.9f                          \
                list_remove_head.9f                     \
                list_remove_tail.9f                     \
                list_tail.9f                            \
                makecom_g0.9f                           \
                makecom_g0_s.9f                         \
                makecom_g1.9f                           \
                makecom_g5.9f                           \
                membar_consumer.9f                      \
                membar_enter.9f                         \
                membar_exit.9f                          \
                membar_producer.9f                      \
                memcmp.9f                               \
                memcpy.9f                               \
                memmove.9f                              \
                memset.9f                               \
                minphys.9f                              \
                mod_info.9f                             \
                mod_modname.9f                          \
                mod_remove.9f                           \
                mutex_destroy.9f                        \
                mutex_enter.9f                          \
                mutex_exit.9f                           \
                mutex_init.9f                           \
                mutex_owned.9f                          \
                mutex_tryenter.9f                       \
                net_event_notify_unregister.9f          \
                net_instance_notify_unregister.9f       \
                net_instance_protocol_unregister.9f     \
                numtos.9f                               \
                nv_alloc_fini.9f                        \
                nv_alloc_init.9f                        \
                nvlist_add_boolean_array.9f             \
                nvlist_add_boolean_value.9f             \
                nvlist_add_byte.9f                      \
                nvlist_add_byte_array.9f                \
                nvlist_add_int16.9f                     \
                nvlist_add_int16_array.9f               \
                nvlist_add_int32.9f                     \
                nvlist_add_int32_array.9f               \
                nvlist_add_int64.9f                     \
                nvlist_add_int64_array.9f               \
                nvlist_add_int8.9f                      \
                nvlist_add_int8_array.9f                \
                nvlist_add_nvlist.9f                    \
                nvlist_add_nvlist_array.9f              \
                nvlist_add_nvpair.9f                    \
                nvlist_add_string.9f                    \
                nvlist_add_string_array.9f              \
                nvlist_add_uint16.9f                    \
                nvlist_add_uint16_array.9f              \
                nvlist_add_uint32.9f                    \
                nvlist_add_uint32_array.9f              \
                nvlist_add_uint64.9f                    \
                nvlist_add_uint64_array.9f              \
                nvlist_add_uint8.9f                     \
                nvlist_add_uint8_array.9f               \
                nvlist_dup.9f                           \
                nvlist_exists.9f                        \
```



```
                nvlist_free.9f                          \
                nvlist_lookup_boolean_array.9f          \
                nvlist_lookup_boolean_value.9f          \
                nvlist_lookup_byte.9f                   \
                nvlist_lookup_byte_array.9f             \
                nvlist_lookup_int16.9f                  \
                nvlist_lookup_int16_array.9f            \
                nvlist_lookup_int32.9f                  \
                nvlist_lookup_int32_array.9f            \
                nvlist_lookup_int64.9f                  \
                nvlist_lookup_int64_array.9f            \
                nvlist_lookup_int8.9f                   \
                nvlist_lookup_int8_array.9f             \
                nvlist_lookup_nvlist.9f                 \
                nvlist_lookup_nvlist_array.9f           \
                nvlist_lookup_pairs.9f                  \
                nvlist_lookup_string.9f                 \
                nvlist_lookup_string_array.9f           \
                nvlist_lookup_uint16.9f                 \
                nvlist_lookup_uint16_array.9f           \
                nvlist_lookup_uint32.9f                 \
                nvlist_lookup_uint32_array.9f           \
                nvlist_lookup_uint64.9f                 \
                nvlist_lookup_uint64_array.9f           \
                nvlist_lookup_uint8.9f                  \
                nvlist_lookup_uint8_array.9f            \
                nvlist_merge.9f                         \
                nvlist_pack.9f                          \
                nvlist_remove_all.9f                    \
                nvlist_size.9f                          \
                nvlist_t.9f                             \
                nvlist_unpack.9f                        \
                nvlist_xalloc.9f                        \
                nvlist_xdup.9f                          \
                nvlist_xpack.9f                         \
                nvlist_xunpack.9f                       \
                nvpair_name.9f                          \
                nvpair_type.9f                          \
                nvpair_value_boolean_array.9f           \
                nvpair_value_byte_array.9f              \
                nvpair_value_int16.9f                   \
                nvpair_value_int16_array.9f             \
                nvpair_value_int32.9f                   \
                nvpair_value_int32_array.9f             \
                nvpair_value_int64.9f                   \
                nvpair_value_int64_array.9f             \
                nvpair_value_int8.9f                    \
                nvpair_value_int8_array.9f              \
                nvpair_value_nvlist.9f                  \
                nvpair_value_nvlist_array.9f            \
                nvpair_value_string.9f                  \
                nvpair_value_string_array.9f            \
                nvpair_value_uint16.9f                  \
                nvpair_value_uint16_array.9f            \
                nvpair_value_uint32.9f                  \
                nvpair_value_uint32_array.9f            \
                nvpair_value_uint64.9f                  \
                nvpair_value_uint64_array.9f            \
                nvpair_value_uint8.9f                   \
                nvpair_value_uint8_array.9f             \
                otherq.9f                               \
                outl.9f                                 \
                outw.9f                                 \
                pci_config_get16.9f                     \
                pci_config_get32.9f                     \
                pci_config_get64.9f                     \
```

```
1050                    pci_config_getb.9f                      \
1051                    pci_config_getl.9f                      \
1052                    pci_config_getll.9f                     \
1053                    pci_config_getw.9f                      \
1054                    pci_config_put16.9f                     \
1055                    pci_config_put32.9f                     \
1056                    pci_config_put64.9f                     \
1057                    pci_config_put8.9f                      \
1058                    pci_config_putb.9f                      \
1059                    pci_config_putl.9f                      \
1060                    pci_config_putll.9f                     \
1061                    pci_config_putw.9f                      \
1062                    pci_config_teardown.9f                  \
1063                    pci_ereport_post.9f                     \
1064                    pci_ereport_teardown.9f                 \
1065                    pci_restore_config_regs.9f              \
1066                    pm_idle_component.9f                    \
1067                    pm_lower_power.9f                       \
1068                    priv_policy_choice.9f                   \
1069                    priv_policy_only.9f                     \
1070                    proc_ref.9f                             \
1071                    proc_unref.9f                           \
1072                    qprocsoff.9f                            \
1073                    qwait_sig.9f                            \
1074                    rd.9f                                   \
1075                    repinsb.9f                              \
1076                    repinsd.9f                              \
1077                    repinsw.9f                              \
1078                    repoutsb.9f                             \
1079                    repoutsd.9f                             \
1080                    repoutsw.9f                             \
1081                    rmallocmap_wait.9f                      \
1082                    rmfreemap.9f                            \
1083                    rw_destroy.9f                           \
1084                    rw_downgrade.9f                         \
1085                    rw_enter.9f                             \
1086                    rw_exit.9f                              \
1087                    rw_init.9f                              \
1088                    rw_read_locked.9f                       \
1089                    rw_tryenter.9f                          \
1090                    rw_tryupgrade.9f                        \
1091                    samestr.9f                              \
1092                    scsi_dmafree.9f                         \
1093                    scsi_dname.9f                           \
1093                    scsi_hba_attach.9f                      \
1094                    scsi_hba_detach.9f                      \
1095                    scsi_hba_fini.9f                        \
1096                    scsi_hba_pkt_free.9f                    \
1097                    scsi_hba_tran_free.9f                   \
1098                    scsi_ifsetcap.9f                        \
1099                    scsi_mname.9f                           \
1100                    scsi_pktfree.9f                         \
1101                    scsi_resalloc.9f                        \
1102                    scsi_resfree.9f                         \
1103                    scsi_rname.9f                           \
1104                    scsi_sense_asc.9f                       \
1105                    scsi_sense_ascq.9f                      \
1106                    scsi_sense_cmdspecific_uint64.9f        \
1107                    scsi_sense_info_uint64.9f               \
1108                    scsi_sname.9f                           \
1109                    scsi_unslave.9f                         \
1110                    sema_destroy.9f                         \
1111                    sema_init.9f                            \
1112                    sema_p.9f                               \
1113                    sema_p_sig.9f                           \
1114                    sema_tryp.9f                            \
```



```
1115                    sema_v.9f                               \
1116                    strcasecmp.9f                           \
1117                    strchr.9f                               \
1118                    strcmp.9f                               \
1119                    strcpy.9f                               \
1120                    strdup.9f                               \
1121                    strfree.9f                              \
1122                    strlcat.9f                              \
1123                    strlcpy.9f                              \
1124                    strlen.9f                               \
1125                    strncasecmp.9f                          \
1126                    strncat.9f                              \
1127                    strncmp.9f                              \
1128                    strncpy.9f                              \
1129                    strnlen.9f                              \
1130                    strrchr.9f                              \
1131                    strspn.9f                               \
1132                    taskq_suspended.9f                      \
1133                    uconv_u16tou8.9f                        \
1134                    uconv_u32tou16.9f                       \
1135                    uconv_u32tou8.9f                        \
1136                    uconv_u8tou16.9f                        \
1137                    uconv_u8tou32.9f                        \
1138                    unfreezestr.9f                          \
1139                    usb_alloc_bulk_req.9f                   \
1140                    usb_alloc_ctrl_req.9f                   \
1141                    usb_alloc_intr_req.9f                   \
1142                    usb_alloc_isoc_req.9f                   \
1143                    usb_client_detach.9f                    \
1144                    usb_free_bulk_req.9f                    \
1145                    usb_free_ctrl_req.9f                    \
1146                    usb_free_descr_tree.9f                  \
1147                    usb_free_dev_data.9f                    \
1148                    usb_free_intr_req.9f                    \
1149                    usb_free_isoc_req.9f                    \
1150                    usb_get_if_number.9f                    \
1151                    usb_owns_device.9f                      \
1152                    usb_pipe_ctrl_xfer_wait.9f              \
1153                    usb_pipe_get_private.9f                 \
1154                    usb_pipe_stop_intr_polling.9f           \
1155                    usb_pipe_stop_isoc_polling.9f           \
1156                    usb_print_descr_tree.9f                 \
1157                    usb_set_alt_if.9f                       \
1158                    usb_set_cfg.9f                          \
1159                    usb_unregister_hotplug_cbs.9f           \
1160                    va_copy.9f                              \
1161                    va_end.9f                               \
1162                    va_start.9f                             \
1163                    vcmn_err.9f                             \
1164                    wr.9f                                   \
1165                    zcmn_err.9f

1167 assert.9f                          := LINKSRC = ASSERT.9f

1169 intro.9f                           := LINKSRC = Intro.9f

1171 otherq.9f                          := LINKSRC = OTHERQ.9f

1173 rd.9f                             := LINKSRC = RD.9f

1175 samestr.9f                        := LINKSRC = SAMESTR.9f

1177 SIZEOF_PTR.9f                     := LINKSRC = STRUCT_DECL.9f
1178 SIZEOF_STRUCT.9f                  := LINKSRC = STRUCT_DECL.9f
1179 STRUCT_BUF.9f                     := LINKSRC = STRUCT_DECL.9f
1180 STRUCT_FADDR.9f                   := LINKSRC = STRUCT_DECL.9f
```

```
1181 STRUCT_FGET.9f                     := LINKSRC = STRUCT_DECL.9f
1182 STRUCT_FGETP.9f                    := LINKSRC = STRUCT_DECL.9f
1183 STRUCT_FSET.9f                     := LINKSRC = STRUCT_DECL.9f
1184 STRUCT_FSETP.9f                    := LINKSRC = STRUCT_DECL.9f
1185 STRUCT_HANDLE.9f                   := LINKSRC = STRUCT_DECL.9f
1186 STRUCT_INIT.9f                     := LINKSRC = STRUCT_DECL.9f
1187 STRUCT_SET_HANDLE.9f               := LINKSRC = STRUCT_DECL.9f
1188 STRUCT_SIZE.9f                     := LINKSRC = STRUCT_DECL.9f

1190 wr.9f                              := LINKSRC = WR.9f

1192 atomic_add_16.9f                   := LINKSRC = atomic_add.9f
1193 atomic_add_16_nv.9f                := LINKSRC = atomic_add.9f
1194 atomic_add_32.9f                   := LINKSRC = atomic_add.9f
1195 atomic_add_32_nv.9f                := LINKSRC = atomic_add.9f
1196 atomic_add_64.9f                   := LINKSRC = atomic_add.9f
1197 atomic_add_64_nv.9f                := LINKSRC = atomic_add.9f
1198 atomic_add_8.9f                    := LINKSRC = atomic_add.9f
1199 atomic_add_8_nv.9f                 := LINKSRC = atomic_add.9f
1200 atomic_add_char.9f                 := LINKSRC = atomic_add.9f
1201 atomic_add_char_nv.9f              := LINKSRC = atomic_add.9f
1202 atomic_add_int.9f                  := LINKSRC = atomic_add.9f
1203 atomic_add_int_nv.9f               := LINKSRC = atomic_add.9f
1204 atomic_add_long.9f                 := LINKSRC = atomic_add.9f
1205 atomic_add_long_nv.9f              := LINKSRC = atomic_add.9f
1206 atomic_add_ptr.9f                  := LINKSRC = atomic_add.9f
1207 atomic_add_ptr_nv.9f               := LINKSRC = atomic_add.9f
1208 atomic_add_short.9f                := LINKSRC = atomic_add.9f
1209 atomic_add_short_nv.9f             := LINKSRC = atomic_add.9f
1210 atomic_and_16.9f                   := LINKSRC = atomic_and.9f
1211 atomic_and_16_nv.9f                := LINKSRC = atomic_and.9f
1212 atomic_and_32.9f                   := LINKSRC = atomic_and.9f
1213 atomic_and_32_nv.9f                := LINKSRC = atomic_and.9f
1214 atomic_and_64.9f                   := LINKSRC = atomic_and.9f
1215 atomic_and_64_nv.9f                := LINKSRC = atomic_and.9f
1216 atomic_and_8.9f                    := LINKSRC = atomic_and.9f
1217 atomic_and_8_nv.9f                 := LINKSRC = atomic_and.9f
1218 atomic_and_uchar.9f                := LINKSRC = atomic_and.9f
1219 atomic_and_uchar_nv.9f             := LINKSRC = atomic_and.9f
1220 atomic_and_uint.9f                 := LINKSRC = atomic_and.9f
1221 atomic_and_uint_nv.9f              := LINKSRC = atomic_and.9f
1222 atomic_and_ulong.9f                := LINKSRC = atomic_and.9f
1223 atomic_and_ulong_nv.9f             := LINKSRC = atomic_and.9f
1224 atomic_and_ushort.9f               := LINKSRC = atomic_and.9f
1225 atomic_and_ushort_nv.9f            := LINKSRC = atomic_and.9f

1227 atomic_clear_long_excl.9f          := LINKSRC = atomic_bits.9f
1228 atomic_set_long_excl.9f            := LINKSRC = atomic_bits.9f

1230 atomic_cas_16.9f                   := LINKSRC = atomic_cas.9f
1231 atomic_cas_32.9f                   := LINKSRC = atomic_cas.9f
1232 atomic_cas_64.9f                   := LINKSRC = atomic_cas.9f
1233 atomic_cas_8.9f                    := LINKSRC = atomic_cas.9f
1234 atomic_cas_ptr.9f                  := LINKSRC = atomic_cas.9f
1235 atomic_cas_uchar.9f                := LINKSRC = atomic_cas.9f
1236 atomic_cas_uint.9f                 := LINKSRC = atomic_cas.9f
1237 atomic_cas_ulong.9f                := LINKSRC = atomic_cas.9f
1238 atomic_cas_ushort.9f               := LINKSRC = atomic_cas.9f

1240 atomic_dec_16.9f                   := LINKSRC = atomic_dec.9f
1241 atomic_dec_16_nv.9f                := LINKSRC = atomic_dec.9f
1242 atomic_dec_32.9f                   := LINKSRC = atomic_dec.9f
1243 atomic_dec_32_nv.9f                := LINKSRC = atomic_dec.9f
1244 atomic_dec_64.9f                   := LINKSRC = atomic_dec.9f
1245 atomic_dec_64_nv.9f                := LINKSRC = atomic_dec.9f
1246 atomic_dec_8.9f                    := LINKSRC = atomic_dec.9f
```



```
1247 atomic_dec_8_nv.9f                := LINKSRC = atomic_dec.9f
1248 atomic_dec_ptr.9f                 := LINKSRC = atomic_dec.9f
1249 atomic_dec_ptr_nv.9f              := LINKSRC = atomic_dec.9f
1250 atomic_dec_uchar.9f               := LINKSRC = atomic_dec.9f
1251 atomic_dec_uchar_nv.9f            := LINKSRC = atomic_dec.9f
1252 atomic_dec_uint.9f                := LINKSRC = atomic_dec.9f
1253 atomic_dec_uint_nv.9f             := LINKSRC = atomic_dec.9f
1254 atomic_dec_ulong.9f               := LINKSRC = atomic_dec.9f
1255 atomic_dec_ulong_nv.9f            := LINKSRC = atomic_dec.9f
1256 atomic_dec_ushort.9f              := LINKSRC = atomic_dec.9f
1257 atomic_dec_ushort_nv.9f           := LINKSRC = atomic_dec.9f

1259 atomic_inc_16.9f                   := LINKSRC = atomic_inc.9f
1260 atomic_inc_16_nv.9f                := LINKSRC = atomic_inc.9f
1261 atomic_inc_32.9f                   := LINKSRC = atomic_inc.9f
1262 atomic_inc_32_nv.9f                := LINKSRC = atomic_inc.9f
1263 atomic_inc_64.9f                   := LINKSRC = atomic_inc.9f
1264 atomic_inc_64_nv.9f                := LINKSRC = atomic_inc.9f
1265 atomic_inc_8.9f                    := LINKSRC = atomic_inc.9f
1266 atomic_inc_8_nv.9f                 := LINKSRC = atomic_inc.9f
1267 atomic_inc_ptr.9f                  := LINKSRC = atomic_inc.9f
1268 atomic_inc_ptr_nv.9f               := LINKSRC = atomic_inc.9f
1269 atomic_inc_uchar.9f                := LINKSRC = atomic_inc.9f
1270 atomic_inc_uchar_nv.9f             := LINKSRC = atomic_inc.9f
1271 atomic_inc_uint.9f                 := LINKSRC = atomic_inc.9f
1272 atomic_inc_uint_nv.9f              := LINKSRC = atomic_inc.9f
1273 atomic_inc_ulong.9f                := LINKSRC = atomic_inc.9f
1274 atomic_inc_ulong_nv.9f             := LINKSRC = atomic_inc.9f
1275 atomic_inc_ushort.9f               := LINKSRC = atomic_inc.9f
1276 atomic_inc_ushort_nv.9f            := LINKSRC = atomic_inc.9f

1278 atomic_or_16.9f                    := LINKSRC = atomic_or.9f
1279 atomic_or_16_nv.9f                 := LINKSRC = atomic_or.9f
1280 atomic_or_32.9f                    := LINKSRC = atomic_or.9f
1281 atomic_or_32_nv.9f                 := LINKSRC = atomic_or.9f
1282 atomic_or_64.9f                    := LINKSRC = atomic_or.9f
1283 atomic_or_64_nv.9f                 := LINKSRC = atomic_or.9f
1284 atomic_or_8.9f                     := LINKSRC = atomic_or.9f
1285 atomic_or_8_nv.9f                  := LINKSRC = atomic_or.9f
1286 atomic_or_uchar.9f                 := LINKSRC = atomic_or.9f
1287 atomic_or_uchar_nv.9f              := LINKSRC = atomic_or.9f
1288 atomic_or_uint.9f                  := LINKSRC = atomic_or.9f
1289 atomic_or_uint_nv.9f               := LINKSRC = atomic_or.9f
1290 atomic_or_ulong.9f                 := LINKSRC = atomic_or.9f
1291 atomic_or_ulong_nv.9f              := LINKSRC = atomic_or.9f
1292 atomic_or_ushort.9f                := LINKSRC = atomic_or.9f
1293 atomic_or_ushort_nv.9f             := LINKSRC = atomic_or.9f

1295 atomic_swap_16.9f                  := LINKSRC = atomic_swap.9f
1296 atomic_swap_32.9f                  := LINKSRC = atomic_swap.9f
1297 atomic_swap_64.9f                  := LINKSRC = atomic_swap.9f
1298 atomic_swap_8.9f                   := LINKSRC = atomic_swap.9f
1299 atomic_swap_ptr.9f                 := LINKSRC = atomic_swap.9f
1300 atomic_swap_uchar.9f               := LINKSRC = atomic_swap.9f
1301 atomic_swap_uint.9f                := LINKSRC = atomic_swap.9f
1302 atomic_swap_ulong.9f               := LINKSRC = atomic_swap.9f
1303 atomic_swap_ushort.9f              := LINKSRC = atomic_swap.9f

1305 vcmn_err.9f                        := LINKSRC = cmn_err.9f
1306 zcmn_err.9f                        := LINKSRC = cmn_err.9f

1308 cv_broadcast.9f                   := LINKSRC = condvar.9f
1309 cv_destroy.9f                     := LINKSRC = condvar.9f
1310 cv_init.9f                        := LINKSRC = condvar.9f
1311 cv_reltimedwait.9f                := LINKSRC = condvar.9f
1312 cv_reltimedwait_sig.9f            := LINKSRC = condvar.9f
```

```
cv_signal.9f                        := LINKSRC = condvar.9f
cv_timedwait.9f                     := LINKSRC = condvar.9f
cv_timedwait_sig.9f                 := LINKSRC = condvar.9f
cv_wait.9f                          := LINKSRC = condvar.9f
cv_wait_sig.9f                      := LINKSRC = condvar.9f

csx_Get16.9f                        := LINKSRC = csx_Get8.9f
csx_Get32.9f                        := LINKSRC = csx_Get8.9f
csx_Get64.9f                        := LINKSRC = csx_Get8.9f

csx_GetNextClient.9f                := LINKSRC = csx_GetFirstClient.9f

csx_GetNextTuple.9f                 := LINKSRC = csx_GetFirstTuple.9f

csx_RemoveDeviceNode.9f             := LINKSRC = csx_MakeDeviceNode.9f

csx_Parse_CISTPL_DEVICE_A.9f        := LINKSRC = csx_Parse_CISTPL_DEVICE.9f
csx_Parse_CISTPL_DEVICE_OA.9f       := LINKSRC = csx_Parse_CISTPL_DEVICE.9f
csx_Parse_CISTPL_DEVICE_OC.9f       := LINKSRC = csx_Parse_CISTPL_DEVICE.9f

csx_Parse_CISTPL_JEDEC_A.9f         := LINKSRC = csx_Parse_CISTPL_JEDEC_C.9f

csx_Parse_CISTPL_LONGLINK_C.9f      := LINKSRC = csx_Parse_CISTPL_LONGLINK_A

csx_Put16.9f                        := LINKSRC = csx_Put8.9f
csx_Put32.9f                        := LINKSRC = csx_Put8.9f
csx_Put64.9f                        := LINKSRC = csx_Put8.9f

csx_RepGet16.9f                     := LINKSRC = csx_RepGet8.9f
csx_RepGet32.9f                     := LINKSRC = csx_RepGet8.9f
csx_RepGet64.9f                     := LINKSRC = csx_RepGet8.9f

csx_RepPut16.9f                     := LINKSRC = csx_RepPut8.9f
csx_RepPut32.9f                     := LINKSRC = csx_RepPut8.9f
csx_RepPut64.9f                     := LINKSRC = csx_RepPut8.9f

csx_ReleaseIO.9f                    := LINKSRC = csx_RequestIO.9f

csx_ReleaseIRQ.9f                   := LINKSRC = csx_RequestIRQ.9f

csx_ReleaseSocketMask.9f            := LINKSRC = csx_RequestSocketMask.9f

csx_ReleaseWindow.9f                := LINKSRC = csx_RequestWindow.9f

csx_GetEventMask.9f                 := LINKSRC = csx_SetEventMask.9f

ddi_get_iblock_cookie.9f            := LINKSRC = ddi_add_intr.9f
ddi_remove_intr.9f                  := LINKSRC = ddi_add_intr.9f

ddi_get_soft_iblock_cookie.9f       := LINKSRC = ddi_add_softintr.9f
ddi_remove_softintr.9f              := LINKSRC = ddi_add_softintr.9f
ddi_trigger_softintr.9f             := LINKSRC = ddi_add_softintr.9f

ddi_get_name.9f                     := LINKSRC = ddi_binding_name.9f

ddi_btopr.9f                        := LINKSRC = ddi_btop.9f
ddi_ptob.9f                         := LINKSRC = ddi_btop.9f

ddi_cb_unregister.9f                := LINKSRC = ddi_cb_register.9f

ddi_check_dma_handle.9f             := LINKSRC = ddi_check_acc_handle.9f

crgetgid.9f                         := LINKSRC = ddi_cred.9f
crgetgroups.9f                      := LINKSRC = ddi_cred.9f
crgetngroups.9f                     := LINKSRC = ddi_cred.9f
crgetrgid.9f                        := LINKSRC = ddi_cred.9f
```



```
crgetruid.9f                        := LINKSRC = ddi_cred.9f
crgetsgid.9f                        := LINKSRC = ddi_cred.9f
crgetsuid.9f                        := LINKSRC = ddi_cred.9f
crgetuid.9f                         := LINKSRC = ddi_cred.9f
crgetzoneid.9f                      := LINKSRC = ddi_cred.9f

ddi_devid_free.9f                   := LINKSRC = ddi_devid_compare.9f
ddi_devid_get.9f                    := LINKSRC = ddi_devid_compare.9f
ddi_devid_init.9f                   := LINKSRC = ddi_devid_compare.9f
ddi_devid_register.9f               := LINKSRC = ddi_devid_compare.9f
ddi_devid_sizeof.9f                 := LINKSRC = ddi_devid_compare.9f
ddi_devid_str_decode.9f             := LINKSRC = ddi_devid_compare.9f
ddi_devid_str_encode.9f             := LINKSRC = ddi_devid_compare.9f
ddi_devid_str_free.9f               := LINKSRC = ddi_devid_compare.9f
ddi_devid_unregister.9f             := LINKSRC = ddi_devid_compare.9f
ddi_devid_valid.9f                  := LINKSRC = ddi_devid_compare.9f

ddi_dmae_1stparty.9f                := LINKSRC = ddi_dmae.9f
ddi_dmae_alloc.9f                   := LINKSRC = ddi_dmae.9f
ddi_dmae_disable.9f                 := LINKSRC = ddi_dmae.9f
ddi_dmae_enable.9f                  := LINKSRC = ddi_dmae.9f
ddi_dmae_getattr.9f                 := LINKSRC = ddi_dmae.9f
ddi_dmae_getcnt.9f                  := LINKSRC = ddi_dmae.9f
ddi_dmae_getlim.9f                  := LINKSRC = ddi_dmae.9f
ddi_dmae_prog.9f                    := LINKSRC = ddi_dmae.9f
ddi_dmae_release.9f                 := LINKSRC = ddi_dmae.9f
ddi_dmae_stop.9f                    := LINKSRC = ddi_dmae.9f

ddi_exit_critical.9f                := LINKSRC = ddi_enter_critical.9f

ddi_fls.9f                          := LINKSRC = ddi_ffs.9f

ddi_fm_dma_err_clear.9f             := LINKSRC = ddi_fm_acc_err_clear.9f

ddi_fm_dma_err_get.9f               := LINKSRC = ddi_fm_acc_err_get.9f

ddi_fm_handler_unregister.9f        := LINKSRC = ddi_fm_handler_register.9f

ddi_fm_capable.9f                   := LINKSRC = ddi_fm_init.9f
ddi_fm_fini.9f                      := LINKSRC = ddi_fm_init.9f

ddi_get16.9f                        := LINKSRC = ddi_get8.9f
ddi_get32.9f                        := LINKSRC = ddi_get8.9f
ddi_get64.9f                        := LINKSRC = ddi_get8.9f
ddi_getb.9f                         := LINKSRC = ddi_get8.9f
ddi_getl.9f                         := LINKSRC = ddi_get8.9f
ddi_getll.9f                        := LINKSRC = ddi_get8.9f
ddi_getw.9f                         := LINKSRC = ddi_get8.9f

ddi_set_driver_private.9f           := LINKSRC = ddi_get_driver_private.9f

ddi_get_lbolt64.9f                  := LINKSRC = ddi_get_lbolt.9f

ddi_intr_remove_handler.9f          := LINKSRC = ddi_intr_add_handler.9f

ddi_intr_get_softint_pri.9f         := LINKSRC = ddi_intr_add_softint.9f
ddi_intr_remove_softint.9f          := LINKSRC = ddi_intr_add_softint.9f
ddi_intr_set_softint_pri.9f         := LINKSRC = ddi_intr_add_softint.9f
ddi_intr_trigger_softint.9f         := LINKSRC = ddi_intr_add_softint.9f

ddi_intr_free.9f                    := LINKSRC = ddi_intr_alloc.9f

ddi_intr_block_disable.9f           := LINKSRC = ddi_intr_enable.9f
ddi_intr_block_enable.9f            := LINKSRC = ddi_intr_enable.9f
ddi_intr_disable.9f                 := LINKSRC = ddi_intr_enable.9f
```

```
1445 ddi_intr_set_cap.9f                    := LINKSRC = ddi_intr_get_cap.9f

1447 ddi_intr_get_navail.9f                 := LINKSRC = ddi_intr_get_nintrs.9f

1449 ddi_intr_set_pri.9f                    := LINKSRC = ddi_intr_get_pri.9f

1451 ddi_intr_clr_mask.9f                   := LINKSRC = ddi_intr_set_mask.9f

1453 ddi_io_get16.9f                        := LINKSRC = ddi_io_get8.9f
1454 ddi_io_get32.9f                        := LINKSRC = ddi_io_get8.9f
1455 ddi_io_getb.9f                         := LINKSRC = ddi_io_get8.9f
1456 ddi_io_getl.9f                         := LINKSRC = ddi_io_get8.9f
1457 ddi_io_getw.9f                         := LINKSRC = ddi_io_get8.9f

1459 ddi_io_put16.9f                        := LINKSRC = ddi_io_put8.9f
1460 ddi_io_put32.9f                        := LINKSRC = ddi_io_put8.9f
1461 ddi_io_putb.9f                         := LINKSRC = ddi_io_put8.9f
1462 ddi_io_putl.9f                         := LINKSRC = ddi_io_put8.9f
1463 ddi_io_putw.9f                         := LINKSRC = ddi_io_put8.9f

1465 ddi_io_rep_get16.9f                    := LINKSRC = ddi_io_rep_get8.9f
1466 ddi_io_rep_get32.9f                    := LINKSRC = ddi_io_rep_get8.9f
1467 ddi_io_rep_getb.9f                     := LINKSRC = ddi_io_rep_get8.9f
1468 ddi_io_rep_getl.9f                     := LINKSRC = ddi_io_rep_get8.9f
1469 ddi_io_rep_getw.9f                     := LINKSRC = ddi_io_rep_get8.9f

1471 ddi_io_rep_put16.9f                    := LINKSRC = ddi_io_rep_put8.9f
1472 ddi_io_rep_put32.9f                    := LINKSRC = ddi_io_rep_put8.9f
1473 ddi_io_rep_putb.9f                     := LINKSRC = ddi_io_rep_put8.9f
1474 ddi_io_rep_putl.9f                     := LINKSRC = ddi_io_rep_put8.9f
1475 ddi_io_rep_putw.9f                     := LINKSRC = ddi_io_rep_put8.9f

1477 ddi_unmap_regs.9f                      := LINKSRC = ddi_map_regs.9f

1479 ddi_mem_get16.9f                       := LINKSRC = ddi_mem_get8.9f
1480 ddi_mem_get32.9f                       := LINKSRC = ddi_mem_get8.9f
1481 ddi_mem_get64.9f                       := LINKSRC = ddi_mem_get8.9f
1482 ddi_mem_getb.9f                        := LINKSRC = ddi_mem_get8.9f
1483 ddi_mem_getl.9f                        := LINKSRC = ddi_mem_get8.9f
1484 ddi_mem_getll.9f                       := LINKSRC = ddi_mem_get8.9f
1485 ddi_mem_getw.9f                        := LINKSRC = ddi_mem_get8.9f

1487 ddi_mem_put16.9f                       := LINKSRC = ddi_mem_put8.9f
1488 ddi_mem_put32.9f                       := LINKSRC = ddi_mem_put8.9f
1489 ddi_mem_put64.9f                       := LINKSRC = ddi_mem_put8.9f
1490 ddi_mem_putb.9f                        := LINKSRC = ddi_mem_put8.9f
1491 ddi_mem_putl.9f                        := LINKSRC = ddi_mem_put8.9f
1492 ddi_mem_putll.9f                       := LINKSRC = ddi_mem_put8.9f
1493 ddi_mem_putw.9f                        := LINKSRC = ddi_mem_put8.9f

1495 ddi_mem_rep_get16.9f                   := LINKSRC = ddi_mem_rep_get8.9f
1496 ddi_mem_rep_get32.9f                   := LINKSRC = ddi_mem_rep_get8.9f
1497 ddi_mem_rep_get64.9f                   := LINKSRC = ddi_mem_rep_get8.9f
1498 ddi_mem_rep_getb.9f                    := LINKSRC = ddi_mem_rep_get8.9f
1499 ddi_mem_rep_getl.9f                    := LINKSRC = ddi_mem_rep_get8.9f
1500 ddi_mem_rep_getll.9f                   := LINKSRC = ddi_mem_rep_get8.9f
1501 ddi_mem_rep_getw.9f                    := LINKSRC = ddi_mem_rep_get8.9f

1503 ddi_mem_rep_put16.9f                   := LINKSRC = ddi_mem_rep_put8.9f
1504 ddi_mem_rep_put32.9f                   := LINKSRC = ddi_mem_rep_put8.9f
1505 ddi_mem_rep_put64.9f                   := LINKSRC = ddi_mem_rep_put8.9f
1506 ddi_mem_rep_putb.9f                    := LINKSRC = ddi_mem_rep_put8.9f
1507 ddi_mem_rep_putl.9f                    := LINKSRC = ddi_mem_rep_put8.9f
1508 ddi_mem_rep_putll.9f                   := LINKSRC = ddi_mem_rep_put8.9f
1509 ddi_mem_rep_putw.9f                    := LINKSRC = ddi_mem_rep_put8.9f
```



```
1511 ddi_modclose.9f                        := LINKSRC = ddi_modopen.9f
1512 ddi_modsym.9f                         := LINKSRC = ddi_modopen.9f

1514 ddi_peek16.9f                         := LINKSRC = ddi_peek.9f
1515 ddi_peek32.9f                         := LINKSRC = ddi_peek.9f
1516 ddi_peek64.9f                         := LINKSRC = ddi_peek.9f
1517 ddi_peek8.9f                          := LINKSRC = ddi_peek.9f
1518 ddi_peekc.9f                          := LINKSRC = ddi_peek.9f
1519 ddi_peekd.9f                          := LINKSRC = ddi_peek.9f
1520 ddi_peekl.9f                          := LINKSRC = ddi_peek.9f
1521 ddi_peeks.9f                          := LINKSRC = ddi_peek.9f

1523 ddi_poke16.9f                         := LINKSRC = ddi_poke.9f
1524 ddi_poke32.9f                         := LINKSRC = ddi_poke.9f
1525 ddi_poke64.9f                         := LINKSRC = ddi_poke.9f
1526 ddi_poke8.9f                          := LINKSRC = ddi_poke.9f
1527 ddi_pokec.9f                          := LINKSRC = ddi_poke.9f
1528 ddi_poked.9f                          := LINKSRC = ddi_poke.9f
1529 ddi_pokel.9f                          := LINKSRC = ddi_poke.9f
1530 ddi_pokes.9f                          := LINKSRC = ddi_poke.9f

1532 ddi_prop_modify.9f                    := LINKSRC = ddi_prop_create.9f
1533 ddi_prop_remove.9f                    := LINKSRC = ddi_prop_create.9f
1534 ddi_prop_remove_all.9f                := LINKSRC = ddi_prop_create.9f
1535 ddi_prop_undefine.9f                  := LINKSRC = ddi_prop_create.9f

1537 ddi_prop_get_int64.9f                 := LINKSRC = ddi_prop_get_int.9f

1539 ddi_prop_free.9f                      := LINKSRC = ddi_prop_lookup.9f
1540 ddi_prop_lookup_byte_array.9f         := LINKSRC = ddi_prop_lookup.9f
1541 ddi_prop_lookup_int64_array.9f        := LINKSRC = ddi_prop_lookup.9f
1542 ddi_prop_lookup_int_array.9f          := LINKSRC = ddi_prop_lookup.9f
1543 ddi_prop_lookup_string.9f             := LINKSRC = ddi_prop_lookup.9f
1544 ddi_prop_lookup_string_array.9f       := LINKSRC = ddi_prop_lookup.9f

1546 ddi_getlongprop.9f                    := LINKSRC = ddi_prop_op.9f
1547 ddi_getlongprop_buf.9f                := LINKSRC = ddi_prop_op.9f
1548 ddi_getprop.9f                        := LINKSRC = ddi_prop_op.9f
1549 ddi_getproplen.9f                     := LINKSRC = ddi_prop_op.9f

1551 ddi_prop_update_byte_array.9f         := LINKSRC = ddi_prop_update.9f
1552 ddi_prop_update_int.9f                := LINKSRC = ddi_prop_update.9f
1553 ddi_prop_update_int64.9f              := LINKSRC = ddi_prop_update.9f
1554 ddi_prop_update_int64_array.9f        := LINKSRC = ddi_prop_update.9f
1555 ddi_prop_update_int_array.9f          := LINKSRC = ddi_prop_update.9f
1556 ddi_prop_update_string.9f             := LINKSRC = ddi_prop_update.9f
1557 ddi_prop_update_string_array.9f       := LINKSRC = ddi_prop_update.9f

1559 ddi_put16.9f                          := LINKSRC = ddi_put8.9f
1560 ddi_put32.9f                          := LINKSRC = ddi_put8.9f
1561 ddi_put64.9f                          := LINKSRC = ddi_put8.9f
1562 ddi_putb.9f                           := LINKSRC = ddi_put8.9f
1563 ddi_putl.9f                           := LINKSRC = ddi_put8.9f
1564 ddi_putll.9f                          := LINKSRC = ddi_put8.9f
1565 ddi_putw.9f                           := LINKSRC = ddi_put8.9f

1567 ddi_rep_get16.9f                      := LINKSRC = ddi_rep_get8.9f
1568 ddi_rep_get32.9f                      := LINKSRC = ddi_rep_get8.9f
1569 ddi_rep_get64.9f                      := LINKSRC = ddi_rep_get8.9f
1570 ddi_rep_getb.9f                       := LINKSRC = ddi_rep_get8.9f
1571 ddi_rep_getl.9f                       := LINKSRC = ddi_rep_get8.9f
1572 ddi_rep_getll.9f                      := LINKSRC = ddi_rep_get8.9f
1573 ddi_rep_getw.9f                       := LINKSRC = ddi_rep_get8.9f

1575 ddi_rep_put16.9f                      := LINKSRC = ddi_rep_put8.9f
1576 ddi_rep_put32.9f                      := LINKSRC = ddi_rep_put8.9f
```

```
1577 ddi_rep_put64.9f                    := LINKSRC = ddi_rep_put8.9f
1578 ddi_rep_putb.9f                     := LINKSRC = ddi_rep_put8.9f
1579 ddi_rep_putl.9f                     := LINKSRC = ddi_rep_put8.9f
1580 ddi_rep_putll.9f                    := LINKSRC = ddi_rep_put8.9f
1581 ddi_rep_putw.9f                     := LINKSRC = ddi_rep_put8.9f

1583 ddi_segmap_setup.9f                 := LINKSRC = ddi_segmap.9f

1585 ddi_get_soft_state.9f               := LINKSRC = ddi_soft_state.9f
1586 ddi_soft_state_fini.9f              := LINKSRC = ddi_soft_state.9f
1587 ddi_soft_state_free.9f              := LINKSRC = ddi_soft_state.9f
1588 ddi_soft_state_init.9f              := LINKSRC = ddi_soft_state.9f
1589 ddi_soft_state_zalloc.9f            := LINKSRC = ddi_soft_state.9f

1591 ddi_strtoull.9f                     := LINKSRC = ddi_strtoll.9f

1593 ddi_umem_free.9f                    := LINKSRC = ddi_umem_alloc.9f

1595 ddi_umem_unlock.9f                  := LINKSRC = ddi_umem_lock.9f

1597 devmap_umem_setup.9f                := LINKSRC = devmap_devmem_setup.9f

1599 ddi_devmap_segmap.9f                := LINKSRC = devmap_setup.9f

1601 devmap_load.9f                      := LINKSRC = devmap_unload.9f

1603 dlerrorack.9f                       := LINKSRC = dlbindack.9f
1604 dlokack.9f                          := LINKSRC = dlbindack.9f
1605 dlphysaddrack.9f                    := LINKSRC = dlbindack.9f
1606 dluderrorind.9f                     := LINKSRC = dlbindack.9f

1608 desballoc.9f                       := LINKSRC = esballoc.9f

1610 unfreezestr.9f                     := LINKSRC = freezestr.9f

1612 free_pktiopb.9f                    := LINKSRC = get_pktiopb.9f

1614 gld_intr.9f                        := LINKSRC = gld.9f
1615 gld_mac_alloc.9f                   := LINKSRC = gld.9f
1616 gld_mac_free.9f                    := LINKSRC = gld.9f
1617 gld_recv.9f                        := LINKSRC = gld.9f
1618 gld_register.9f                    := LINKSRC = gld.9f
1619 gld_sched.9f                       := LINKSRC = gld.9f
1620 gld_unregister.9f                  := LINKSRC = gld.9f

1622 id32_free.9f                       := LINKSRC = id32_alloc.9f
1623 id32_lookup.9f                     := LINKSRC = id32_alloc.9f

1625 inl.9f                            := LINKSRC = inb.9f
1626 inw.9f                            := LINKSRC = inb.9f
1627 repinsb.9f                        := LINKSRC = inb.9f
1628 repinsd.9f                        := LINKSRC = inb.9f
1629 repinsw.9f                        := LINKSRC = inb.9f

1631 kmem_free.9f                       := LINKSRC = kmem_alloc.9f
1632 kmem_zalloc.9f                     := LINKSRC = kmem_alloc.9f

1634 kmem_cache_alloc.9f                := LINKSRC = kmem_cache_create.9f
1635 kmem_cache_destroy.9f              := LINKSRC = kmem_cache_create.9f
1636 kmem_cache_free.9f                 := LINKSRC = kmem_cache_create.9f
1637 kmem_cache_set_move.9f             := LINKSRC = kmem_cache_create.9f

1639 kstat_named_setstr.9f              := LINKSRC = kstat_named_init.9f

1641 kstat_runq_back_to_waitq.9f        := LINKSRC = kstat_queue.9f
1642 kstat_runq_enter.9f                := LINKSRC = kstat_queue.9f
```



```
1643 kstat_runq_exit.9f                 := LINKSRC = kstat_queue.9f
1644 kstat_waitq_enter.9f              := LINKSRC = kstat_queue.9f
1645 kstat_waitq_exit.9f               := LINKSRC = kstat_queue.9f
1646 kstat_waitq_to_runq.9f            := LINKSRC = kstat_queue.9f

1648 ldi_awrite.9f                     := LINKSRC = ldi_aread.9f

1650 ldi_get_devid.9f                  := LINKSRC = ldi_get_dev.9f
1651 ldi_get_minor_name.9f            := LINKSRC = ldi_get_dev.9f
1652 ldi_get_otyp.9f                  := LINKSRC = ldi_get_dev.9f

1654 ldi_ident_from_dip.9f            := LINKSRC = ldi_ident_from_dev.9f
1655 ldi_ident_from_stream.9f         := LINKSRC = ldi_ident_from_dev.9f
1656 ldi_ident_release.9f             := LINKSRC = ldi_ident_from_dev.9f

1658 ldi_close.9f                     := LINKSRC = ldi_open_by_dev.9f
1659 ldi_open_by_devid.9f             := LINKSRC = ldi_open_by_dev.9f
1660 ldi_open_by_name.9f              := LINKSRC = ldi_open_by_dev.9f

1662 ldi_prop_get_int64.9f            := LINKSRC = ldi_prop_get_int.9f

1664 ldi_prop_lookup_byte_array.9f    := LINKSRC = ldi_prop_lookup_int_array.9
1665 ldi_prop_lookup_int64_array.9f   := LINKSRC = ldi_prop_lookup_int_array.9
1666 ldi_prop_lookup_string.9f        := LINKSRC = ldi_prop_lookup_int_array.9
1667 ldi_prop_lookup_string_array.9f  := LINKSRC = ldi_prop_lookup_int_array.9

1669 ldi_getmsg.9f                    := LINKSRC = ldi_putmsg.9f

1671 ldi_write.9f                     := LINKSRC = ldi_read.9f

1673 list_destroy.9f                  := LINKSRC = list_create.9f
1674 list_head.9f                     := LINKSRC = list_create.9f
1675 list_insert_after.9f             := LINKSRC = list_create.9f
1676 list_insert_before.9f            := LINKSRC = list_create.9f
1677 list_insert_head.9f              := LINKSRC = list_create.9f
1678 list_insert_tail.9f              := LINKSRC = list_create.9f
1679 list_is_empty.9f                 := LINKSRC = list_create.9f
1680 list_link_active.9f              := LINKSRC = list_create.9f
1681 list_link_init.9f                := LINKSRC = list_create.9f
1682 list_link_replace.9f             := LINKSRC = list_create.9f
1683 list_move_tail.9f                := LINKSRC = list_create.9f
1684 list_next.9f                     := LINKSRC = list_create.9f
1685 list_prev.9f                     := LINKSRC = list_create.9f
1686 list_remove.9f                   := LINKSRC = list_create.9f
1687 list_remove_head.9f              := LINKSRC = list_create.9f
1688 list_remove_tail.9f              := LINKSRC = list_create.9f
1689 list_tail.9f                     := LINKSRC = list_create.9f

1691 makecom_g0.9f                    := LINKSRC = makecom.9f
1692 makecom_g0_s.9f                  := LINKSRC = makecom.9f
1693 makecom_g1.9f                    := LINKSRC = makecom.9f
1694 makecom_g5.9f                    := LINKSRC = makecom.9f

1696 membar_consumer.9f               := LINKSRC = membar_ops.9f
1697 membar_enter.9f                  := LINKSRC = membar_ops.9f
1698 membar_exit.9f                   := LINKSRC = membar_ops.9f
1699 membar_producer.9f               := LINKSRC = membar_ops.9f

1701 memcmp.9f                        := LINKSRC = memchr.9f
1702 memcpy.9f                        := LINKSRC = memchr.9f
1703 memmove.9f                       := LINKSRC = memchr.9f
1704 memset.9f                        := LINKSRC = memchr.9f

1706 mod_info.9f                      := LINKSRC = mod_install.9f
1707 mod_modname.9f                   := LINKSRC = mod_install.9f
1708 mod_remove.9f                    := LINKSRC = mod_install.9f
```

```
mutex_destroy.9f                        := LINKSRC = mutex.9f
mutex_enter.9f                          := LINKSRC = mutex.9f
mutex_exit.9f                           := LINKSRC = mutex.9f
mutex_init.9f                           := LINKSRC = mutex.9f
mutex_owned.9f                          := LINKSRC = mutex.9f
mutex_tryenter.9f                       := LINKSRC = mutex.9f

net_event_notify_unregister.9f          := LINKSRC = net_event_notify_register.9

net_instance_notify_unregister.9f       := LINKSRC = net_instance_notify_registe

net_instance_protocol_unregister.9f     := LINKSRC = net_protocol_notify_registe

nvlist_add_boolean_array.9f             := LINKSRC = nvlist_add_boolean.9f
nvlist_add_boolean_value.9f             := LINKSRC = nvlist_add_boolean.9f
nvlist_add_byte.9f                      := LINKSRC = nvlist_add_boolean.9f
nvlist_add_byte_array.9f                := LINKSRC = nvlist_add_boolean.9f
nvlist_add_int16.9f                     := LINKSRC = nvlist_add_boolean.9f
nvlist_add_int16_array.9f               := LINKSRC = nvlist_add_boolean.9f
nvlist_add_int32.9f                     := LINKSRC = nvlist_add_boolean.9f
nvlist_add_int32_array.9f               := LINKSRC = nvlist_add_boolean.9f
nvlist_add_int64.9f                     := LINKSRC = nvlist_add_boolean.9f
nvlist_add_int64_array.9f               := LINKSRC = nvlist_add_boolean.9f
nvlist_add_int8.9f                      := LINKSRC = nvlist_add_boolean.9f
nvlist_add_int8_array.9f                := LINKSRC = nvlist_add_boolean.9f
nvlist_add_nvlist.9f                    := LINKSRC = nvlist_add_boolean.9f
nvlist_add_nvlist_array.9f              := LINKSRC = nvlist_add_boolean.9f
nvlist_add_nvpair.9f                    := LINKSRC = nvlist_add_boolean.9f
nvlist_add_string.9f                    := LINKSRC = nvlist_add_boolean.9f
nvlist_add_string_array.9f              := LINKSRC = nvlist_add_boolean.9f
nvlist_add_uint16.9f                    := LINKSRC = nvlist_add_boolean.9f
nvlist_add_uint16_array.9f              := LINKSRC = nvlist_add_boolean.9f
nvlist_add_uint32.9f                    := LINKSRC = nvlist_add_boolean.9f
nvlist_add_uint32_array.9f              := LINKSRC = nvlist_add_boolean.9f
nvlist_add_uint64.9f                    := LINKSRC = nvlist_add_boolean.9f
nvlist_add_uint64_array.9f              := LINKSRC = nvlist_add_boolean.9f
nvlist_add_uint8.9f                     := LINKSRC = nvlist_add_boolean.9f
nvlist_add_uint8_array.9f               := LINKSRC = nvlist_add_boolean.9f
nvlist_t.9f                             := LINKSRC = nvlist_add_boolean.9f

nv_alloc_fini.9f                        := LINKSRC = nvlist_alloc.9f
nv_alloc_init.9f                        := LINKSRC = nvlist_alloc.9f
nvlist_dup.9f                           := LINKSRC = nvlist_alloc.9f
nvlist_free.9f                          := LINKSRC = nvlist_alloc.9f
nvlist_merge.9f                         := LINKSRC = nvlist_alloc.9f
nvlist_pack.9f                          := LINKSRC = nvlist_alloc.9f
nvlist_size.9f                          := LINKSRC = nvlist_alloc.9f
nvlist_unpack.9f                        := LINKSRC = nvlist_alloc.9f
nvlist_xalloc.9f                        := LINKSRC = nvlist_alloc.9f
nvlist_xdup.9f                          := LINKSRC = nvlist_alloc.9f
nvlist_xpack.9f                         := LINKSRC = nvlist_alloc.9f
nvlist_xunpack.9f                       := LINKSRC = nvlist_alloc.9f

nvlist_lookup_boolean_array.9f          := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_boolean_value.9f          := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_byte.9f                   := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_byte_array.9f             := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_int16.9f                  := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_int16_array.9f            := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_int32.9f                  := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_int32_array.9f            := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_int64.9f                  := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_int64_array.9f            := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_int8.9f                   := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_int8_array.9f             := LINKSRC = nvlist_lookup_boolean.9f
```



```
nvlist_lookup_nvlist.9f                 := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_nvlist_array.9f           := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_pairs.9f                  := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_string.9f                 := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_string_array.9f           := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_uint16.9f                 := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_uint16_array.9f           := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_uint32.9f                 := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_uint32_array.9f           := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_uint64.9f                 := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_uint64_array.9f           := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_uint8.9f                  := LINKSRC = nvlist_lookup_boolean.9f
nvlist_lookup_uint8_array.9f            := LINKSRC = nvlist_lookup_boolean.9f

nvlist_exists.9f                        := LINKSRC = nvlist_lookup_nvpair.9f

nvpair_name.9f                          := LINKSRC = nvlist_next_nvpair.9f
nvpair_type.9f                          := LINKSRC = nvlist_next_nvpair.9f

nvlist_remove_all.9f                    := LINKSRC = nvlist_remove.9f

nvpair_value_boolean_array.9f           := LINKSRC = nvpair_value_byte.9f
nvpair_value_byte_array.9f              := LINKSRC = nvpair_value_byte.9f
nvpair_value_int16.9f                   := LINKSRC = nvpair_value_byte.9f
nvpair_value_int16_array.9f             := LINKSRC = nvpair_value_byte.9f
nvpair_value_int32.9f                   := LINKSRC = nvpair_value_byte.9f
nvpair_value_int32_array.9f             := LINKSRC = nvpair_value_byte.9f
nvpair_value_int64.9f                   := LINKSRC = nvpair_value_byte.9f
nvpair_value_int64_array.9f             := LINKSRC = nvpair_value_byte.9f
nvpair_value_int8.9f                    := LINKSRC = nvpair_value_byte.9f
nvpair_value_int8_array.9f              := LINKSRC = nvpair_value_byte.9f
nvpair_value_nvlist.9f                  := LINKSRC = nvpair_value_byte.9f
nvpair_value_nvlist_array.9f            := LINKSRC = nvpair_value_byte.9f
nvpair_value_string.9f                  := LINKSRC = nvpair_value_byte.9f
nvpair_value_string_array.9f            := LINKSRC = nvpair_value_byte.9f
nvpair_value_uint16.9f                  := LINKSRC = nvpair_value_byte.9f
nvpair_value_uint16_array.9f            := LINKSRC = nvpair_value_byte.9f
nvpair_value_uint32.9f                  := LINKSRC = nvpair_value_byte.9f
nvpair_value_uint32_array.9f            := LINKSRC = nvpair_value_byte.9f
nvpair_value_uint64.9f                  := LINKSRC = nvpair_value_byte.9f
nvpair_value_uint64_array.9f            := LINKSRC = nvpair_value_byte.9f
nvpair_value_uint8.9f                   := LINKSRC = nvpair_value_byte.9f
nvpair_value_uint8_array.9f             := LINKSRC = nvpair_value_byte.9f

outl.9f                                 := LINKSRC = outb.9f
outw.9f                                 := LINKSRC = outb.9f
repoutsb.9f                             := LINKSRC = outb.9f
repoutsd.9f                             := LINKSRC = outb.9f
repoutsw.9f                             := LINKSRC = outb.9f

pci_config_get16.9f                     := LINKSRC = pci_config_get8.9f
pci_config_get32.9f                     := LINKSRC = pci_config_get8.9f
pci_config_get64.9f                     := LINKSRC = pci_config_get8.9f
pci_config_getb.9f                      := LINKSRC = pci_config_get8.9f
pci_config_getl.9f                      := LINKSRC = pci_config_get8.9f
pci_config_getll.9f                     := LINKSRC = pci_config_get8.9f
pci_config_getw.9f                      := LINKSRC = pci_config_get8.9f
pci_config_put16.9f                     := LINKSRC = pci_config_get8.9f
pci_config_put32.9f                     := LINKSRC = pci_config_get8.9f
pci_config_put64.9f                     := LINKSRC = pci_config_get8.9f
pci_config_put8.9f                      := LINKSRC = pci_config_get8.9f
pci_config_putb.9f                      := LINKSRC = pci_config_get8.9f
pci_config_putl.9f                      := LINKSRC = pci_config_get8.9f
pci_config_putll.9f                     := LINKSRC = pci_config_get8.9f
pci_config_putw.9f                      := LINKSRC = pci_config_get8.9f
```

```
1841 pci_config_teardown.9f                  := LINKSRC = pci_config_setup.9f

1843 pci_ereport_post.9f                     := LINKSRC = pci_ereport_setup.9f
1844 pci_ereport_teardown.9f                 := LINKSRC = pci_ereport_setup.9f

1846 pci_restore_config_regs.9f               := LINKSRC = pci_save_config_regs.9f

1848 minphys.9f                              := LINKSRC = physio.9f

1850 pm_idle_component.9f                    := LINKSRC = pm_busy_component.9f

1852 pm_lower_power.9f                       := LINKSRC = pm_raise_power.9f

1854 priv_policy_choice.9f                   := LINKSRC = priv_policy.9f
1855 priv_policy_only.9f                     := LINKSRC = priv_policy.9f

1857 proc_ref.9f                             := LINKSRC = proc_signal.9f
1858 proc_unref.9f                           := LINKSRC = proc_signal.9f

1860 qprocsoff.9f                            := LINKSRC = qprocson.9f

1862 qwait_sig.9f                            := LINKSRC = qwait.9f

1864 rmallocmap_wait.9f                      := LINKSRC = rmallocmap.9f
1865 rmfreemap.9f                           := LINKSRC = rmallocmap.9f

1867 rw_destroy.9f                           := LINKSRC = rwlock.9f
1868 rw_downgrade.9f                        := LINKSRC = rwlock.9f
1869 rw_enter.9f                            := LINKSRC = rwlock.9f
1870 rw_exit.9f                             := LINKSRC = rwlock.9f
1871 rw_init.9f                             := LINKSRC = rwlock.9f
1872 rw_read_locked.9f                      := LINKSRC = rwlock.9f
1873 rw_tryenter.9f                         := LINKSRC = rwlock.9f
1874 rw_tryupgrade.9f                       := LINKSRC = rwlock.9f

1876 scsi_dname.9f                          := LINKSRC = scsi_cname.9f
1877 scsi_mname.9f                          := LINKSRC = scsi_cname.9f
1878 scsi_rname.9f                          := LINKSRC = scsi_cname.9f
1879 scsi_sname.9f                          := LINKSRC = scsi_cname.9f

1881 scsi_dmafree.9f                        := LINKSRC = scsi_dmaget.9f

1883 scsi_sense_cmdspecific_uint64.9f       := LINKSRC = scsi_ext_sense_fields.9f
1884 scsi_sense_info_uint64.9f              := LINKSRC = scsi_ext_sense_fields.9f

1886 scsi_hba_attach.9f                     := LINKSRC = scsi_hba_attach_setup.9f
1886 scsi_hba_detach.9f                     := LINKSRC = scsi_hba_attach_setup.9f

1888 scsi_hba_fini.9f                       := LINKSRC = scsi_hba_init.9f

1890 scsi_hba_pkt_free.9f                   := LINKSRC = scsi_hba_pkt_alloc.9f

1892 scsi_hba_tran_free.9f                  := LINKSRC = scsi_hba_tran_alloc.9f

1894 scsi_ifsetcap.9f                       := LINKSRC = scsi_ifgetcap.9f

1896 scsi_pktfree.9f                        := LINKSRC = scsi_pktalloc.9f
1897 scsi_resalloc.9f                       := LINKSRC = scsi_pktalloc.9f
1898 scsi_resfree.9f                        := LINKSRC = scsi_pktalloc.9f

1900 scsi_sense_asc.9f                      := LINKSRC = scsi_sense_key.9f
1901 scsi_sense_ascq.9f                     := LINKSRC = scsi_sense_key.9f

1903 scsi_unslave.9f                        := LINKSRC = scsi_unprobe.9f

1905 sema_destroy.9f                       := LINKSRC = semaphore.9f
```



```
1906 sema_init.9f                          := LINKSRC = semaphore.9f
1907 sema_p.9f                            := LINKSRC = semaphore.9f
1908 sema_p_sig.9f                        := LINKSRC = semaphore.9f
1909 sema_tryp.9f                         := LINKSRC = semaphore.9f
1910 sema_v.9f                            := LINKSRC = semaphore.9f

1912 numtos.9f                            := LINKSRC = stoi.9f

1914 ddi_strdup.9f                        := LINKSRC = string.9f
1915 strcasecmp.9f                        := LINKSRC = string.9f
1916 strchr.9f                            := LINKSRC = string.9f
1917 strcmp.9f                            := LINKSRC = string.9f
1918 strcpy.9f                            := LINKSRC = string.9f
1919 strdup.9f                            := LINKSRC = string.9f
1920 strfree.9f                           := LINKSRC = string.9f
1921 strlcat.9f                           := LINKSRC = string.9f
1922 strlcpy.9f                           := LINKSRC = string.9f
1923 strlen.9f                            := LINKSRC = string.9f
1924 strncasecmp.9f                       := LINKSRC = string.9f
1925 strncat.9f                           := LINKSRC = string.9f
1926 strncmp.9f                           := LINKSRC = string.9f
1927 strncpy.9f                           := LINKSRC = string.9f
1928 strnlen.9f                           := LINKSRC = string.9f
1929 strrchr.9f                           := LINKSRC = string.9f
1930 strspn.9f                            := LINKSRC = string.9f

1932 ddi_taskq_create.9f                   := LINKSRC = taskq.9f
1933 ddi_taskq_destroy.9f                  := LINKSRC = taskq.9f
1934 ddi_taskq_dispatch.9f                 := LINKSRC = taskq.9f
1935 ddi_taskq_resume.9f                   := LINKSRC = taskq.9f
1936 ddi_taskq_suspend.9f                  := LINKSRC = taskq.9f
1937 ddi_taskq_wait.9f                     := LINKSRC = taskq.9f
1938 taskq_suspended.9f                    := LINKSRC = taskq.9f

1940 uconv_u16tou8.9f                      := LINKSRC = uconv_u16tou32.9f
1941 uconv_u32tou16.9f                     := LINKSRC = uconv_u16tou32.9f
1942 uconv_u32tou8.9f                      := LINKSRC = uconv_u16tou32.9f
1943 uconv_u8tou16.9f                      := LINKSRC = uconv_u16tou32.9f
1944 uconv_u8tou32.9f                      := LINKSRC = uconv_u16tou32.9f

1946 usb_alloc_bulk_req.9f                := LINKSRC = usb_alloc_request.9f
1947 usb_alloc_ctrl_req.9f                := LINKSRC = usb_alloc_request.9f
1948 usb_alloc_intr_req.9f                := LINKSRC = usb_alloc_request.9f
1949 usb_alloc_isoc_req.9f                := LINKSRC = usb_alloc_request.9f
1950 usb_free_bulk_req.9f                 := LINKSRC = usb_alloc_request.9f
1951 usb_free_ctrl_req.9f                 := LINKSRC = usb_alloc_request.9f
1952 usb_free_intr_req.9f                 := LINKSRC = usb_alloc_request.9f
1953 usb_free_isoc_req.9f                 := LINKSRC = usb_alloc_request.9f
1954 usb_client_detach.9f                 := LINKSRC = usb_client_attach.9f

1956 usb_get_if_number.9f                 := LINKSRC = usb_get_alt_if.9f
1957 usb_owns_device.9f                   := LINKSRC = usb_get_alt_if.9f
1958 usb_set_alt_if.9f                    := LINKSRC = usb_get_alt_if.9f

1960 usb_set_cfg.9f                       := LINKSRC = usb_get_cfg.9f

1962 usb_free_descr_tree.9f               := LINKSRC = usb_get_dev_data.9f
1963 usb_free_dev_data.9f                 := LINKSRC = usb_get_dev_data.9f
1964 usb_print_descr_tree.9f              := LINKSRC = usb_get_dev_data.9f

1966 usb_pipe_ctrl_xfer_wait.9f           := LINKSRC = usb_pipe_ctrl_xfer.9f
1967 usb_pipe_stop_intr_polling.9f        := LINKSRC = usb_pipe_intr_xfer.9f
1968 usb_pipe_stop_isoc_polling.9f        := LINKSRC = usb_pipe_isoc_xfer.9f

1970 usb_pipe_get_private.9f              := LINKSRC = usb_pipe_set_private.9f
```

```
1972 usb_unregister_hotplug_cbs.9f           := LINKSRC = usb_register_hotplug_cbs.9f

1974 va_copy.9f                              := LINKSRC = va_arg.9f
1975 va_end.9f                               := LINKSRC = va_arg.9f
1976 va_start.9f                             := LINKSRC = va_arg.9f

1978 .KEEP_STATE:

1980 include         $(SRC)/man/Makefile.man

1982 install:        $(ROOTMANFILES) $(ROOTMANLINKS)
```

```
**********************************************************
    2783 Wed Feb 26 14:23:23 2014
new/usr/src/man/man9f/ddi_iomin.9f
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
'\" te
.\" Copyright (c) 2006, Sun Microsystems, Inc.
.\" Copyright 2014 Pluribus Networks, Inc.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH DDI_IOMIN 9F "Feb 26, 2014"
.TH DDI_IOMIN 9F "Apr 04, 2006"
.SH NAME
ddi_iomin \- find minimum alignment and transfer size for DMA
.SH SYNOPSIS
.LP
.nf
#include <sys/conf.h>
#include <sys/ddi.h>
#include <sys/sunddi.h>



\fBint\fR \fBddi_iomin\fR(\fBdev_info_t *\fR\fIdip\fR, \fBint\fR \fIinitial\fR,
.fi

.SH INTERFACE LEVEL
.sp
.LP
Solaris DDI specific (Solaris DDI). This interface is obsolete.
.SH PARAMETERS
.sp
.ne 2
.na
\fB\fIdip\fR\fR
.ad
.RS 13n
A pointer to the device's \fBdev_info\fR structure.
.RE

.sp
.ne 2
.na
\fB\fIinitial\fR\fR
.ad
.RS 13n
The initial minimum \fBDMA\fR transfer size in bytes. This may be zero or an
appropriate \fBdlim_minxfer\fR value for the device's \fBddi_dma_attr\fR
structure (see \fBddi_dma_attr\fR(9S)). This value must be a power of two.
The initial minimum \fBDMA\fRtransfer size in bytes. This may be zero or an
appropriate \fBdlim_minxfer\fR value for device's \fBddi_dma_lim\fR structure
(see \fBddi_dma_lim_sparc\fR(9S) or \fBddi_dma_lim_x86\fR(9S)). This value must
be a power of two.
.RE

.sp
.ne 2
.na
\fB\fIstreaming\fR\fR
.ad
.RS 13n
This argument, if non-zero, indicates that the returned value should be
modified to account for \fIstreaming\fR mode accesses (see
\fBddi_dma_req\fR(9S) for a discussion of streaming versus non-streaming access
```



```
mode).
.RE

.SH DESCRIPTION
.sp
.LP
The \fBddi_iomin()\fR function, finds out the minimum \fBDMA\fR transfer size
The \fBddi_iomin()\fR function, finds out the minimum \fBDMA\fRtransfer size
for the device pointed to by \fIdip\fR. This provides a mechanism by which a
driver can determine the effects of underlying caches as well as intervening
bus adapters on the granularity of a DMA transfer.
.SH RETURN VALUES
.sp
.LP
The \fBddi_iomin()\fR function returns the minimum \fBDMA\fRtransfer size for
the calling device, or it returns zero, which means that you cannot get there
from here.
.SH CONTEXT
.sp
.LP
This function can be called from user, interrupt, or kernel context.
.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Interface Stability     Obsolete
.TE

.SH SEE ALSO
.sp
.LP
\fBddi_dma_attr\fR(9S), \fBddi_dma_sync\fR(9F)
\fBddi_dma_devalign\fR(9F), \fBddi_dma_setup\fR(9F), \fBddi_dma_sync\fR(9F),
\fBddi_dma_lim_sparc\fR(9S), \fBddi_dma_lim_x86\fR(9S), \fBddi_dma_req\fR(9S)
.sp
.LP
\fIWriting Device Drivers\fR
```

```
**********************************************************
    5080 Wed Feb 26 14:23:23 2014
new/usr/src/man/man9f/get_pktiopb.9f
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
'\" te
.\" Copyright (c) 2006, Sun Microsystems, Inc., All Rights Reserved
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH GET_PKTIOPB 9F "Feb 26, 2014"
.TH GET_PKTIOPB 9F "Jan 16, 2006"
.SH NAME
get_pktiopb, free_pktiopb \- allocate/free a SCSI packet in the iopb map
.SH SYNOPSIS
.LP
.nf
#include <sys/scsi/scsi.h>



\fBstruct scsi_pkt *\fR\fBget_pktiopb\fR(\fBstruct scsi_address *\fR\fIap\fR,
     \fBcaddr_t *\fR\fIdatap\fR, \fBint\fR \fIcdblen\fR, \fBint\fR \fIstatuslen\
     \fBint\fR \fIreadflag\fR, \fBint (*\fR\fIcallback\fR);
.fi

.LP
.nf
\fBvoid\fR \fBfree_pktiopb\fR(\fBstruct scsi_pkt *\fR\fIpkt\fR, \fBcaddr_t\fR \f
.fi

.SH INTERFACE LEVEL
.sp
.LP
These interfaces are obsolete. Use \fBscsi_alloc_consistent_buf\fR(9F) instead
of \fBget_pktiopb()\fR. Use \fBscsi_free_consistent_buf\fR(9F) instead of
\fBfree_pktiopb()\fR.
.SH PARAMETERS
.sp
.ne 2
.na
\fB\fIap\fR\fR
.ad
.RS 13n
Pointer to the target's \fBscsi_address\fR structure.
.RE

.sp
.ne 2
.na
\fB\fIdatap\fR\fR
.ad
.RS 13n
Pointer to the address of the packet, set by this function.
.RE

.sp
.ne 2
.na
\fB\fIcdblen\fR\fR
.ad
.RS 13n
Number of bytes required for the \fBSCSI \fRcommand descriptor block (CDB).
.RE
```



```
.sp
.ne 2
.na
\fB\fIstatuslen\fR\fR
.ad
.RS 13n
Number of bytes required for the \fBSCSI \fRstatus area.
.RE

.sp
.ne 2
.na
\fB\fIdatalen\fR\fR
.ad
.RS 13n
Number of bytes required for the data area of the \fBSCSI \fRcommand.
.RE

.sp
.ne 2
.na
\fB\fIreadflag\fR\fR
.ad
.RS 13n
If non-zero, data will be transferred from the \fBSCSI \fRtarget.
.RE

.sp
.ne 2
.na
\fB\fIcallback\fR\fR
.ad
.RS 13n
Pointer to a callback function, or \fBNULL_FUNC\fR or \fBSLEEP_FUNC\fR
.RE

.sp
.ne 2
.na
\fB\fIpkt\fR\fR
.ad
.RS 13n
Pointer to a \fBscsi_pkt\fR(9S) structure.
.RE

.SH DESCRIPTION
.sp
.LP
The \fBget_pktiopb()\fR function allocates a \fBscsi_pkt\fR structure that has
a small data area allocated. It is used by some \fBSCSI \fRcommands such as
\fBREQUEST_SENSE\fR, which involve a small amount of data and require
cache-consistent memory for proper operation. It uses \fBddi_iopb_alloc\fR(9F)
for allocating the data area and \fBscsi_resalloc\fR(9F) to allocate the packet
and \fBDMA\fR resources.
.sp
.LP
\fIcallback\fR indicates what \fBget_pktiopb()\fR should do when resources are
not available:
.sp
.ne 2
.na
\fB\fBNULL_FUNC\fR\fR
.ad
.RS 16n
Do not wait for resources. Return a \fINULL\fR pointer.
.RE
```

```
 127 .sp
 128 .ne 2
 129 .na
 130 \fB\fBSLEEP_FUNC\fR\fR
 131 .ad
 132 .RS 16n
 133 Wait indefinitely for resources.
 134 .RE

 136 .sp
 137 .ne 2
 138 .na
 139 \fBOther Values\fR
 140 .ad
 141 .RS 16n
 142 \fIcallback\fR points to a function which is called when resources may have
 143 become available. \fIcallback\fR \fBmust\fR return either \fB0\fR (indicating
 144 that it attempted to allocate resources but failed to do so again), in which
 145 case it is put back on a list to be called again later, or \fB1\fR indicating
 146 either success in allocating resources or indicating that it no longer cares
 147 for a retry.
 148 .RE

 150 .sp
 151 .LP
 152 The \fBfree_pktiopb()\fR function is used for freeing the packet and its
 153 associated resources.
 154 .SH RETURN VALUES
 155 .sp
 156 .LP
 157 The \fBget_pktiopb()\fR function returns a pointer to the newly allocated
 158 \fBscsi_pkt\fR or a \fINULL\fR pointer.
 159 .SH CONTEXT
 160 .sp
 161 .LP
 162 If \fIcallback\fR is \fBSLEEP_FUNC\fR, then this routine can be called only
 163 from user or kernel context. Otherwise, it can be called from user, interrupt,
 164 or kernel context. The \fIcallback\fR function should not block or call
 165 routines that block.
 166 .sp
 167 .LP
 168 The \fBfree_pktiopb()\fR function can be called from user, interrupt, or kernel
 169 context.
 170 .SH ATTRIBUTES
 171 .sp
 172 .LP
 173 See \fBattributes\fR(5) for a description of the following attributes:
 174 .sp

 176 .sp
 177 .TS
 178 box;
 179 c | c
 180 l | l .
 181 ATTRIBUTE TYPE  ATTRIBUTE VALUE
 182 _
 183 Stability Level Obsolete
 184 .TE

 186 .SH SEE ALSO
 187 .sp
 188 .LP
 189 \fBattributes\fR(5),
 189 \fIattributes\fR(5), \fIddi_iopb_alloc\fR(9F),
 190 \fBscsi_alloc_consistent_buf\fR(9F), \fBscsi_free_consistent_buf\fR(9F),
```



```
 191 \fBscsi_pktalloc\fR(9F), \fBscsi_resalloc\fR(9F), \fBscsi_pkt\fR(9S)
 192 .sp
 193 .LP
 194 \fIWriting Device Drivers\fR
 195 .SH NOTES
 196 .sp
 197 .LP
 198 The \fBget_pktiopb()\fR and \fBfree_pktiopb()\fR functions are obsolete and
 199 will be discontinued in a future release. These functions have been replaced
 200 by, respectively, \fBscsi_alloc_consistent_buf\fR(9F) and
 201 \fBscsi_free_consistent_buf\fR(9F).
 202 .sp
 203 .LP
 204 The \fBget_pktiopb()\fR function uses scarce resources. For this reason and its
 205 obsolescence (see above), its use is discouraged.
```

```
**********************************************************
    7234 Wed Feb 26 14:23:23 2014
new/usr/src/man/man9f/scsi_hba_attach_setup.9f
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
'\" te
.\" Copyright (c) 2006 Sun Microsystems, Inc., All Rights Reserved
.\" Copyright 2014 Pluribus Networks, Inc.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH SCSI_HBA_ATTACH_SETUP 9F "May 30, 2006"
.SH NAME
scsi_hba_attach_setup, scsi_hba_attach, scsi_hba_detach \- SCSI HBA attach and
detach routines
.SH SYNOPSIS
.LP
.nf
#include <sys/scsi/scsi.h>



\fBint\fR \fBscsi_hba_attach_setup\fR(\fBdev_info_t *\fR\fIdip\fR, \fBddi_dma_at
     \fBscsi_hba_tran_t *\fR\fIhba_tran\fR, \fBint\fR \fIhba_flags\fR);
.fi

.LP
.nf
\fBint\fR \fBscsi_hba_attach\fR(\fBdev_info_t *\fR\fIdip\fR, \fBddi_dma_lim_t *\
     \fBscsi_hba_tran_t *\fR\fIhba_tran\fR, \fBint\fR \fIhba_flags\fR, \fBvoid *
.fi

.LP
.nf
\fBint\fR \fBscsi_hba_detach\fR(\fBdev_info_t *\fR\fIdip\fR);
.fi

.SH INTERFACE LEVEL
.sp
.LP
Solaris architecture specific (Solaris DDI).
.SH PARAMETERS
.sp
.ne 2
.na
\fB\fIdip\fR\fR
.ad
.RS 16n
Pointer to the \fBdev_info_t\fR structure that refers to the instance of the
HBA device.
.RE

.sp
.ne 2
.na
\fB\fIhba_lim\fR\fR
.ad
.RS 16n
Pointer to a \fBddi_dma_lim\fR(9S) structure.
.RE

.sp
.ne 2
.na
\fB\fIhba_tran\fR\fR
```



```
.ad
.RS 16n
Pointer to a \fBscsi_hba_tran\fR(9S) structure.
.RE

.sp
.ne 2
.na
\fB\fIhba_flags\fR\fR
.ad
.RS 16n
Flag modifiers. The defined flag values are \fBSCSI_HBA_TRAN_CLONE\fR,
\fBSCSI_HBA_TRAN_SCB\fR, and \fBSCSI_HBA_TRAN_CDB\fR.
.RE

.sp
.ne 2
.na
\fB\fIhba_options\fR\fR
.ad
.RS 16n
Optional features provided by the HBA driver for future extensions; must be
\fINULL\fR.
.RE

.sp
.ne 2
.na
\fB\fIhba_dma_attr\fR\fR
.ad
.RS 16n
Pointer to a \fBddi_dma_attr\fR(9S) structure.
.RE

.SH DESCRIPTION
.sp
.LP
The \fBscsi_hba_attach_setup()\fR function registers the
The \fBscsi_hba_attach_setup()\fR function is the recommended interface over
the \fBscsi_hba_attach()\fR function.
.SS "scsi_hba_attach_setup(\|) scsi_hba_attach(\|)"
.sp
.LP
The \fBscsi_hba_attach()\fR function registers the \fIhba_lim\fR DMA limits and
the \fIhba_tran\fR transport vectors of each instance of the HBA device defined
by \fIdip\fR. The \fBscsi_hba_attach_setup()\fR function registers the
\fIhba_dma_attr\fR DMA attributes and the \fIhba_tran\fR transport vectors of
each instance of the HBA device defined by \fIdip\fR. The HBA driver can pass
different DMA limits or DMA attributes and the transport vectors for each
instance of the device to support any constraints imposed by the  HBA itself.
.sp
.LP
The \fBscsi_hba_attach_setup()\fR function uses the
The \fBscsi_hba_attach()\fR and \fBscsi_hba_attach_setup()\fR functions use the
\fBdev_bus_ops\fR field in the \fBdev_ops\fR(9S) structure. The HBA driver
should initialize this field to \fINULL\fR before calling
\fBscsi_hba_attach_setup()\fR.
\fBscsi_hba_attach()\fR or \fBscsi_hba_attach_setup()\fR.
.sp
.LP
If \fBSCSI_HBA_TRAN_CLONE\fR is requested in \fIhba_flags\fR, the
\fBhba_tran\fR structure is cloned once for each target that is attached to the
HBA. The structure is cloned before the \fBtran_tgt_init\fR(9E) entry point is
called to initialize a target. At all subsequent HBA entry points, including
\fBtran_tgt_init\fR(9E), the \fBscsi_hba_tran_t\fR structure passed as an
argument or found in a \fBscsi_address\fR structure is the cloned
```

\fBscsi_hba_tran_t\fR structure,which allows the HBA to use the
\fBtran_tgt_private\fR field in the \fBscsi_hba_tran_t\fR structure to point to
per-target data. The HBA should free only the same \fBscsi_hba_tran_t\fR
structure allocated when the HBA detaches. All cloned \fBscsi_hba_tran_t\fR
structures that are allocated by the system are freed by the system.
.sp
.LP
The flags \fBSCSI_HBA_TRAN_CDB\fR and \fBSCSI_HBA_TRAN_SCB\fR are only valid
when \fBtran_setup_pkt()\fR is used. See \fBtran_setup_pkt\fR(9E) for
information on using these flags.
.sp
.LP
The \fBscsi_hba_attach_setup()\fR function attaches
The \fBscsi_hba_attach()\fR and \fBscsi_hba_attach_setup()\fR functions attach
a number of integer-valued properties to \fIdip\fR, unless properties of the
same name are already attached to the node. An HBA driver should retrieve these
configuration parameters via \fBddi_prop_get_int\fR(9F), and respect any
settings for features provided the HBA.
.sp
.ne 2
.na
\fB\fBscsi-options\fR\fR
.ad
.RS 26n
\fBOptional\fR \fBSCSI\fR \fBconfiguration bits\fR
.RE

.sp
.ne 2
.na
\fB\fBSCSI_OPTIONS_DR\fR\fR
.ad
.RS 26n
If not set, the HBA should not grant Disconnect privileges to target devices.
.RE

.sp
.ne 2
.na
\fB\fBSCSI_OPTIONS_TAG\fR\fR
.ad
.RS 26n
If not set, the HBA should not operate in Command Tagged Queueing mode.
.RE

.sp
.ne 2
.na
\fB\fBSCSI_OPTIONS_PARITY\fR\fR
.ad
.RS 26n
If not set, the HBA should not operate in parity mode.
.RE

.sp
.ne 2
.na
\fB\fBSCSI_OPTIONS_QAS\fR\fR
.ad
.RS 26n
If not set, the HBA should not make use of the Quick Arbitration Select
feature. Consult your Sun hardware documentation to determine whether your
machine supports QAS.
.RE

.sp

.ne 2
.na
\fB\fBSCSI_OPTIONS_FAST\fR\fR
.ad
.RS 26n
If not set, the HBA should not operate the bus in FAST SCSI mode.
.RE

.sp
.ne 2
.na
\fB\fBSCSI_OPTIONS_FAST20\fR\fR
.ad
.RS 26n
If not set, the HBA should not operate the bus in FAST20 SCSI mode.
.RE

.sp
.ne 2
.na
\fB\fBSCSI_OPTIONS_FAST40\fR\fR
.ad
.RS 26n
If not set, the HBA should not operate the bus in FAST40 SCSI mode.
.RE

.sp
.ne 2
.na
\fB\fBSCSI_OPTIONS_FAST80\fR\fR
.ad
.RS 26n
If not set, the HBA should not operate the bus in FAST80 SCSI mode.
.RE

.sp
.ne 2
.na
\fB\fBSCSI_OPTIONS_FAST160\fR\fR
.ad
.RS 26n
If not set, the HBA should not operate the bus in FAST160 SCSI mode.
.RE

.sp
.ne 2
.na
\fB\fBSCSI_OPTIONS_FAST320\fR\fR
.ad
.RS 26n
If not set, the HBA should not operate the bus in FAST320 SCSI mode.
.RE

.sp
.ne 2
.na
\fB\fBSCSI_OPTIONS_WIDE\fR\fR
.ad
.RS 26n
If not set, the HBA should not operate the bus in WIDE SCSI mode.
.RE

.sp
.ne 2
.na
\fB\fBSCSI_OPTIONS_SYNC\fR\fR

```
.ad
.RS 26n
If not set, the HBA should not operate the bus in synchronous transfer mode.
.RE

.sp
.ne 2
.na
\fB\fBscsi-reset-delay\fR\fR
.ad
.RS 26n
SCSI bus or device reset recovery time, in milliseconds.
.RE

.sp
.ne 2
.na
\fB\fBscsi-selection-timeout\fR\fR
.ad
.RS 26n
Default SCSI selection phase timeout value, in milliseconds. Please refer to
individual HBA man pages for any HBA-specific information
.RE

.SS "scsi_hba_detach(\|)"
.sp
.LP
The \fBscsi_hba_detach()\fR function removes the reference to the DMA limits or
attributes structure and the transport vector for the given instance of an HBA
driver.
.SH RETURN VALUES
.sp
.LP
The \fBscsi_hba_attach_setup()\fR and
The \fBscsi_hba_attach()\fR, \fBscsi_hba_attach_setup()\fR, and
\fBscsi_hba_detach()\fR functions return \fBDDI_SUCCESS\fR if the function call
succeeds, and return \fBDDI_FAILURE\fR on failure.
.SH CONTEXT
.sp
.LP
The \fBscsi_hba_attach_setup()\fR function should
The \fBscsi_hba_attach()\fR and \fBscsi_hba_attach_setup()\fR functions should
be called from \fBattach\fR(9E). The \fBscsi_hba_detach()\fR function should be
called from \fBdetach\fR(9E).
.SH SEE ALSO
.sp
.LP
\fBattach\fR(9E), \fBdetach\fR(9E), \fBtran_setup_pkt\fR(9E),
\fBtran_tgt_init\fR(9E), \fBddi_prop_get_int\fR(9F), \fBddi_dma_attr\fR(9S),
\fBdev_ops\fR(9S), \fBscsi_address\fR(9S),
\fBddi_dma_lim\fR(9S), \fBdev_ops\fR(9S), \fBscsi_address\fR(9S),
\fBscsi_hba_tran\fR(9S)
.sp
.LP
\fIWriting Device Drivers\fR
.SH NOTES
.sp
.LP
It is the HBA driver’s responsibility to ensure that no more transport requests
will be taken on behalf of any SCSI target device driver after
\fBscsi_hba_detach()\fR is called.
.sp
.LP
The \fBscsi_hba_attach()\fR function is obsolete and will be discontinued in a
future release. This function is replaced by \fBscsi_hba_attach_setup()\fR.
```

```
**********************************************************
    4291 Wed Feb 26 14:23:23 2014
new/usr/src/man/man9f/scsi_slave.9f
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
'\" te
.\" Copyright (c) 2002, Sun Microsystems, Inc., All Rights Reserved
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH SCSI_SLAVE 9F "Feb 26, 2014"
.TH SCSI_SLAVE 9F "Sep 27, 2002"
.SH NAME
scsi_slave \- utility for SCSI target drivers to establish the presence of a
target
.SH SYNOPSIS
.LP
.nf
#include <sys/scsi/scsi.h>



\fBint\fR \fBscsi_slave\fR(\fBstruct scsi_device *\fR\fIdevp\fR, \fBint\fR \fB(*
.fi

.SH INTERFACE LEVEL
.sp
.LP
The \fBscsi_slave()\fR function is obsolete. This function has been replaced by
\fBscsi_probe\fR(9F).
.SH PARAMETERS
.sp
.ne 2
.na
\fB\fIdevp\fR \fR
.ad
.RS 13n
Pointer to a \fBscsi_device\fR(9S) structure.
.RE

.sp
.ne 2
.na
\fB\fIcallback\fR \fR
.ad
.RS 13n
Pointer to a callback function, \fBNULL_FUNC\fR or \fBSLEEP_FUNC\fR.
.RE

.SH DESCRIPTION
.sp
.LP
\fBscsi_slave()\fR checks for the presence of a \fBSCSI\fR device. Target
drivers may use this function in their \fBprobe\fR(9E) routines.
\fBscsi_slave()\fR determines if the device is present by using a Test Unit
Ready command followed by an Inquiry command. If \fBscsi_slave()\fR is
successful, it will fill in the \fBscsi_inquiry\fR structure, which is the
\fBsd_inq\fR member of the \fBscsi_device\fR(9S) structure, and return
\fBSCSI_PROBE_EXISTS\fR. This information can be used to determine if the
target driver has probed the correct SCSI device type. \fIcallback\fR indicates
what the allocator routines should do when \fBDMA \fRresources are not
available:
.sp
.ne 2
.na
```



```
\fB\fBNULL_FUNC\fR \fR
.ad
.RS 16n
Do not wait for resources. Return a \fINULL\fR pointer.
.RE

.sp
.ne 2
.na
\fB\fBSLEEP_FUNC\fR \fR
.ad
.RS 16n
Wait indefinitely for resources.
.RE

.sp
.ne 2
.na
\fBOther Values\fR
.ad
.RS 16n
\fIcallback\fR points to a function which is called when resources may have
become available. \fIcallback\fR \fBmust\fR return either \fB0\fR (indicating
that it attempted to allocate resources but again failed to do so), in which
case it is put back on a list to be called again later, or \fB1\fR indicating
either success in allocating resources or indicating that it no longer cares
for a retry.
.RE

.SH RETURN VALUES
.sp
.LP
\fBscsi_slave()\fR returns:
.sp
.ne 2
.na
\fB\fBSCSIPROBE_NOMEM\fR \fR
.ad
.RS 22n
No space available for structures.
.RE

.sp
.ne 2
.na
\fB\fBSCSIPROBE_EXISTS\fR \fR
.ad
.RS 22n
Device exists and inquiry data is valid.
.RE

.sp
.ne 2
.na
\fB\fBSCSIPROBE_NONCCS\fR \fR
.ad
.RS 22n
Device exists but inquiry data is not valid.
.RE

.sp
.ne 2
.na
\fB\fBSCSIPROBE_FAILURE\fR \fR
.ad
.RS 22n
```

```
126 Polled command failure.
127 .RE

129 .sp
130 .ne 2
131 .na
132 \fB\fBSCSIPROBE_NORESP\fR \fR
133 .ad
134 .RS 22n
135 No response to \fBTEST UNIT READY\fR.
136 .RE

138 .SH CONTEXT
139 .sp
140 .LP
141 \fBscsi_slave()\fR is normally called from the target driver's \fBprobe\fR(9E)
142 or \fBattach\fR(9E) routine. In any case, this routine should not be called
143 from interrupt context, because it can sleep waiting for memory to be
144 allocated.
145 .SH ATTRIBUTES
146 .sp
147 .LP
148 See \fBattributes\fR(5) for a description of the following attributes:
149 .sp

151 .sp
152 .TS
153 box;
154 c | c
155 l | l .
156 ATTRIBUTE TYPE  ATTRIBUTE VALUE
157 _
158 Stability Level Obsolete
159 .TE

161 .SH SEE ALSO
162 .sp
163 .LP
164 \fBattributes\fR(5), \fBattach\fR(9E), \fBprobe\fR(9E),
165 \fBmakecom\fR(9F), \fBscsi_dmaget\fR(9F),
165 \fBddi_iopb_alloc\fR(9F), \fBmakecom\fR(9F), \fBscsi_dmaget\fR(9F),
166 \fBscsi_ifgetcap\fR(9F), \fBscsi_pktalloc\fR(9F), \fBscsi_poll\fR(9F),
167 \fBscsi_probe\fR(9F), \fBscsi_device\fR(9S)
168 .sp
169 .LP
170  \fIANSI Small Computer System Interface-2 (SCSI-2)\fR
171 .sp
172 .LP
173 \fIWriting Device Drivers\fR
174 .SH NOTES
175 .sp
176 .LP
177 The \fBscsi_slave()\fR function is obsolete and will be discontinued in a
178 future release. This function has been replaced by \fBscsi_probe\fR(9F).
```

```
**********************************************************
    4037 Wed Feb 26 14:23:23 2014
new/usr/src/man/man9s/Intro.9s
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
'\" te
.\" Copyright 2014 Pluribus Networks, Inc.
.\" Copyright (c) 2001, Sun Microsystems, Inc.,  All Rights Reserved.
.\" Copyright 1989 AT&T
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH INTRO 9S "Feb 26, 2014"
.TH INTRO 9S "May 15, 2001"
.SH NAME
Intro, intro \- introduction to kernel data structures and properties
.SH DESCRIPTION
.sp
.LP
Section 9P describes kernel properties used by device drivers. Section 9S
describes the data structures used by drivers to share information between the
driver and the kernel. See \fBIntro\fR(9E) for an overview of device driver
interfaces.
.sp
.LP
In Section 9S, reference pages contain the following headings:
.RS +4
.TP
.ie t \(bu
.el o
\fBNAME\fR summarizes the purpose of the structure or property.
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fBSYNOPSIS\fR lists the include file that defines the structure or property.
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fBINTERFACE\fR \fBLEVEL\fR describes any architecture dependencies.
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fBDESCRIPTION\fR provides general information about the structure or property.
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fBSTRUCTURE\fR \fBMEMBERS\fR lists all accessible structure members (for
Section 9S).
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fBSEE\fR \fBALSO\fR gives sources for further information.
.RE
.sp
.LP
```



```
Of the preceding headings, Section 9P reference pages contain the \fBNAME\fR,
\fBDESCRIPTION\fR, and \fBSEE\fR \fBALSO\fR fields.
.sp
.LP
Every driver MUST include <\fBsys/ddi.h\fR> and <\fBsys/sunddi.h\fR>, in that
order, and as final entries.
.sp
.LP
The following table summarizes the STREAMS structures described in Section 9S.
.sp

.sp
.TS
box;
c | c
l | l .
Structure       Type
_
\fBcopyreq\fR   DDI/DKI
_
\fBcopyresp\fR  DDI/DKI
_
\fBdatab\fR     DDI/DKI
_
\fBfmodsw\fR    Solaris DDI
_
\fBfree_rtn\fR  DDI/DKI
_
\fBiocblk\fR    DDI/DKI
_
\fBlinkblk\fR   DDI/DKI
_
\fBmodule_info\fR       DDI/DKI
_
\fBmsgb\fR      DDI/DKI
_
\fBqband\fR     DDI/DKI
_
\fBqinit\fR     DDI/DKI
_
\fBqueclass\fR  Solaris DDI
_
\fBqueue\fR     DDI/DKI
_
\fBstreamtab\fR DDI/DKI
_
\fBstroptions\fR        DDI/DKI
.TE

.sp
.LP
The following table summarizes structures that are not specific to STREAMS I/O.
.sp

.sp
.TS
box;
c | c
l | l .
Structure       Type
_
\fBaio_req\fR   Solaris DDI
_
\fBbuf\fR       DDI/DKI
_
\fBcb_ops\fR    Solaris DDI
```

```
126 _
127 \fBddi_device_acc_attr\fR       Solaris DDI
128 _
129 \fBddi_dma_attr\fR      Solaris DDI
130 _
131 \fBddi_dma_cookie\fR    Solaris DDI
132 _
132 \fIddi_dma_lim_sparc\fR Solaris SPARC DDI
133 _
134 \fIddi_dma_lim_x86\fR   Solaris x86 DDI
135 _
136 \fIddi_dma_req\fR       Solaris DDI
137 _
133 \fBddi_dmae_req\fR      Solaris x86 DDI
134 _
135 \fBddi_idevice_cookie\fR        Solaris DDI
136 _
137 \fBddi_mapdev_ctl\fR    Solaris DDI
138 _
139 \fBdevmap_callback_ctl\fR       Solaris DDI
140 _
141 \fBdev_ops\fR   Solaris DDI
142 _
143 \fBiovec\fR     DDI/DKI
144 _
145 \fBkstat\fR     Solaris DDI
146 _
147 \fBkstat_intr\fR        Solaris DDI
148 _
149 \fBkstat_io\fR  Solaris DDI
150 _
151 \fBkstat_named\fR       Solaris DDI
152 _
153 \fBmap\fR       DDI/DKI
154 _
155 \fBmodldrv\fR   Solaris DDI
156 _
157 \fBmodlinkage\fR        Solaris DDI
158 _
159 \fBmodlstrmod\fR        Solaris DDI
160 _
161 \fBscsi_address\fR      Solaris DDI
162 _
163 \fBscsi_arq_status\fR   Solaris DDI
164 _
165 \fBscsi_device\fR       Solaris DDI
166 _
167 \fBscsi_extended_sense\fR       Solaris DDI
168 _
169 \fBscsi_hba_tran\fR     Solaris DDI
170 _
171 \fBscsi_inquiry\fR      Solaris DDI
172 _
173 \fBscsi_pkt\fR  Solaris DDI
174 _
175 \fBscsi_status\fR       Solaris DDI
176 _
177 \fBuio\fR       DDI/DKI
178 .TE

180 .SH SEE ALSO
181 .sp
182 .LP
183 \fBIntro\fR(9E)
184 .SH NOTES
185 .sp
```



```
186 .LP
187 Do not declare arrays of structures as the size of the structures can change
188 between releases. Rely only on the structure members listed in this chapter and
189 not on unlisted members or the position of a member in a structure.
```

```
**********************************************************
    2291 Wed Feb 26 14:23:23 2014
new/usr/src/man/man9s/Makefile
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
#
# 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 2011, Richard Lowe
# Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
#

include         $(SRC)/Makefile.master

MANSECT=        9s

MANFILES=       Intro.9s                        \
                aio_req.9s                      \
                buf.9s                          \
                cb_ops.9s                       \
                copyreq.9s                      \
                copyresp.9s                     \
                datab.9s                        \
                ddi_device_acc_attr.9s          \
                ddi_dma_attr.9s                 \
                ddi_dma_cookie.9s               \
                ddi_dma_lim_sparc.9s            \
                ddi_dma_lim_x86.9s              \
                ddi_dma_req.9s                  \
                ddi_dmae_req.9s                 \
                ddi_fm_error.9s                 \
                ddi_idevice_cookie.9s           \
                dev_ops.9s                      \
                devmap_callback_ctl.9s          \
                fmodsw.9s                       \
                free_rtn.9s                     \
                gld_mac_info.9s                 \
                gld_stats.9s                    \
                hook_nic_event.9s               \
                hook_pkt_event.9s               \
                hook_t.9s                       \
                iocblk.9s                       \
                iovec.9s                        \
                kstat.9s                        \
                kstat_intr.9s                   \
                kstat_io.9s                     \
                kstat_named.9s                  \
                linkblk.9s                      \
                modldrv.9s                      \
                modlinkage.9s                   \
                modlmisc.9s                     \
                modlstrmod.9s                   \
                module_info.9s                  \
                msgb.9s                         \
                net_inject_t.9s                 \
                net_instance_t.9s               \
```



```
                qband.9s                        \
                qinit.9s                        \
                queclass.9s                     \
                queue.9s                        \
                scsi_address.9s                 \
                scsi_arq_status.9s              \
                scsi_asc_key_strings.9s         \
                scsi_device.9s                  \
                scsi_extended_sense.9s          \
                scsi_hba_tran.9s                \
                scsi_inquiry.9s                 \
                scsi_pkt.9s                     \
                scsi_status.9s                  \
                streamtab.9s                    \
                stroptions.9s                   \
                tuple.9s                        \
                uio.9s                          \
                usb_bulk_request.9s             \
                usb_callback_flags.9s           \
                usb_cfg_descr.9s                \
                usb_client_dev_data.9s          \
                usb_completion_reason.9s        \
                usb_ctrl_request.9s             \
                usb_dev_descr.9s                \
                usb_dev_qlf_descr.9s            \
                usb_ep_descr.9s                 \
                usb_if_descr.9s                 \
                usb_intr_request.9s             \
                usb_isoc_request.9s             \
                usb_other_speed_cfg_descr.9s    \
                usb_request_attributes.9s       \
                usb_string_descr.9s

MANLINKS=       dblk.9s         \
                ddi_dma_lim.9s  \
                intro.9s        \
                mblk.9s

intro.9s        := LINKSRC = Intro.9s

dblk.9s         := LINKSRC = datab.9s

ddi_dma_lim.9s  := LINKSRC = ddi_dma_lim_sparc.9s

mblk.9s         := LINKSRC = msgb.9s

.KEEP_STATE:

include         $(SRC)/man/Makefile.man

install:        $(ROOTMANFILES) $(ROOTMANLINKS)
```

```
**********************************************************
    3019 Wed Feb 26 14:23:24 2014
new/usr/src/pkg/manifests/system-kernel.man9s.inc
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
```

```
#
# 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 2011, Richard Lowe
# Copyright 2012 Nexenta Systems, Inc. All rights reserved.
#

file path=usr/share/man/man9s/Intro.9s
file path=usr/share/man/man9s/aio_req.9s
file path=usr/share/man/man9s/buf.9s
file path=usr/share/man/man9s/cb_ops.9s
file path=usr/share/man/man9s/copyreq.9s
file path=usr/share/man/man9s/copyresp.9s
file path=usr/share/man/man9s/datab.9s
file path=usr/share/man/man9s/ddi_device_acc_attr.9s
file path=usr/share/man/man9s/ddi_dma_attr.9s
file path=usr/share/man/man9s/ddi_dma_cookie.9s
file path=usr/share/man/man9s/ddi_dma_lim_sparc.9s
file path=usr/share/man/man9s/ddi_dma_lim_x86.9s
file path=usr/share/man/man9s/ddi_dma_req.9s
file path=usr/share/man/man9s/ddi_dmae_req.9s
file path=usr/share/man/man9s/ddi_fm_error.9s
file path=usr/share/man/man9s/ddi_idevice_cookie.9s
file path=usr/share/man/man9s/dev_ops.9s
file path=usr/share/man/man9s/devmap_callback_ctl.9s
file path=usr/share/man/man9s/fmodsw.9s
file path=usr/share/man/man9s/free_rtn.9s
file path=usr/share/man/man9s/gld_mac_info.9s
file path=usr/share/man/man9s/gld_stats.9s
file path=usr/share/man/man9s/hook_nic_event.9s
file path=usr/share/man/man9s/hook_pkt_event.9s
file path=usr/share/man/man9s/hook_t.9s
file path=usr/share/man/man9s/iocblk.9s
file path=usr/share/man/man9s/iovec.9s
file path=usr/share/man/man9s/kstat.9s
file path=usr/share/man/man9s/kstat_intr.9s
file path=usr/share/man/man9s/kstat_io.9s
file path=usr/share/man/man9s/kstat_named.9s
file path=usr/share/man/man9s/linkblk.9s
file path=usr/share/man/man9s/modldrv.9s
file path=usr/share/man/man9s/modlinkage.9s
file path=usr/share/man/man9s/modlmisc.9s
file path=usr/share/man/man9s/modlstrmod.9s
file path=usr/share/man/man9s/module_info.9s
file path=usr/share/man/man9s/msgb.9s
file path=usr/share/man/man9s/net_inject_t.9s
file path=usr/share/man/man9s/net_instance_t.9s
file path=usr/share/man/man9s/qband.9s
file path=usr/share/man/man9s/qinit.9s
file path=usr/share/man/man9s/queclass.9s
file path=usr/share/man/man9s/queue.9s
```



```
file path=usr/share/man/man9s/scsi_address.9s
file path=usr/share/man/man9s/scsi_arq_status.9s
file path=usr/share/man/man9s/scsi_asc_key_strings.9s
file path=usr/share/man/man9s/scsi_device.9s
file path=usr/share/man/man9s/scsi_extended_sense.9s
file path=usr/share/man/man9s/scsi_hba_tran.9s
file path=usr/share/man/man9s/scsi_inquiry.9s
file path=usr/share/man/man9s/scsi_pkt.9s
file path=usr/share/man/man9s/scsi_status.9s
file path=usr/share/man/man9s/streamtab.9s
file path=usr/share/man/man9s/stroptions.9s
file path=usr/share/man/man9s/tuple.9s
file path=usr/share/man/man9s/uio.9s
link path=usr/share/man/man9s/dblk.9s target=datab.9s
link path=usr/share/man/man9s/ddi_dma_lim.9s target=ddi_dma_lim_sparc.9s
link path=usr/share/man/man9s/intro.9s target=Intro.9s
link path=usr/share/man/man9s/mblk.9s target=msgb.9s
```

```
**********************************************************
  182847 Wed Feb 26 14:23:24 2014
new/usr/src/uts/common/io/pcic.c
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
```

```
/*
 * 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.
 */

/*
 * PCIC device/interrupt handler
 *      The "pcic" driver handles the Intel 82365SL, Cirrus Logic
 *      and Toshiba (and possibly other clones) PCMCIA adapter chip
 *      sets.  It implements a subset of Socket Services as defined
 *      in the Solaris PCMCIA design documents
 */

/*
 * currently defined "properties"
 *
 * clock-frequency              bus clock frequency
 * smi                          system management interrupt override
 * need-mult-irq                need status IRQ for each pair of sockets
 * disable-audio                don't route audio signal to speaker
 */


#include <sys/types.h>
#include <sys/inttypes.h>
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/user.h>
#include <sys/buf.h>
#include <sys/file.h>
#include <sys/uio.h>
#include <sys/conf.h>
#include <sys/stat.h>
#include <sys/autoconf.h>
#include <sys/vtoc.h>
#include <sys/dkio.h>
#include <sys/ddi.h>
#include <sys/sunddi.h>
#include <sys/sunndi.h>
```



```
#include <sys/var.h>
#include <sys/callb.h>
#include <sys/open.h>
#include <sys/ddidmareq.h>
#include <sys/dma_engine.h>
#include <sys/kstat.h>
#include <sys/kmem.h>
#include <sys/modctl.h>
#include <sys/pci.h>
#include <sys/pci_impl.h>

#include <sys/pctypes.h>
#include <sys/pcmcia.h>
#include <sys/sservice.h>

#include <sys/note.h>

#include <sys/pcic_reg.h>
#include <sys/pcic_var.h>

#if defined(__i386) || defined(__amd64)
#include <sys/pci_cfgspace.h>
#endif

#if defined(__sparc)
#include <sys/pci/pci_nexus.h>
#endif

#include <sys/hotplug/hpcsvc.h>
#include "cardbus/cardbus.h"

#define SOFTC_SIZE      (sizeof (anp_t))

static int pcic_getinfo(dev_info_t *, ddi_info_cmd_t, void *, void **);
static int pcic_attach(dev_info_t *, ddi_attach_cmd_t);
static int pcic_detach(dev_info_t *, ddi_detach_cmd_t);
static int32_t pcic_quiesce(dev_info_t *);
static uint_t pcic_intr(caddr_t, caddr_t);
static int pcic_do_io_intr(pcicdev_t *, uint32_t);
static int pcic_probe(dev_info_t *);

static int pcic_open(dev_t *, int, int, cred_t *);
static int pcic_close(dev_t, int, int, cred_t *);
static int pcic_ioctl(dev_t, int, intptr_t, int, cred_t *, int *);

typedef struct pcm_regs pcm_regs_t;

static void pcic_init_assigned(dev_info_t *);
static int pcic_apply_avail_ranges(dev_info_t *, pcm_regs_t *,
        pci_regspec_t *, int);
int pci_resource_setup_avail(dev_info_t *, pci_regspec_t *, int);

/*
 * To make this nexus work on x86, we need to have the default ddi_dma_mctl
 * ctlops in the bus_ops structure, just to pass the request to the parent.
 * On x86 platforms the ddi_iobp_alloc(9F) and ddi_mem_alloc(9F) calls
 * are xlated into DMA ctlops. To make this nexus work on x86, we
 * need to have the default ddi_dma_mctl ctlops in the bus_ops
 * structure, just to pass the request to the parent. The correct
 * ctlops should be ddi_no_dma_mctl because so far we don't do DMA.
 */
static
struct bus_ops pcmciabus_ops = {
        BUSO_REV,
        pcmcia_bus_map,
        NULL,
```

```
 123         NULL,
 124         NULL,
 125         i_ddi_map_fault,
 126         ddi_no_dma_map,
 127         ddi_no_dma_allochdl,
 128         ddi_no_dma_freehdl,
 129         ddi_no_dma_bindhdl,
 130         ddi_no_dma_unbindhdl,
 131         ddi_no_dma_flush,
 132         ddi_no_dma_win,
 133         ddi_dma_mctl,
 134         pcmcia_ctlops,
 135         pcmcia_prop_op,
 136         NULL,                           /* (*bus_get_eventcookie)();    */
 137         NULL,                           /* (*bus_add_eventcall)();      */
 138         NULL,                           /* (*bus_remove_eventcall)();   */
 139         NULL,                           /* (*bus_post_event)();         */
 140         NULL,                           /* (*bus_intr_ctl)();           */
 141         NULL,                           /* (*bus_config)();             */
 142         NULL,                           /* (*bus_unconfig)();           */
 143         NULL,                           /* (*bus_fm_init)();            */
 144         NULL,                           /* (*bus_fm_fini)();            */
 145         NULL,                           /* (*bus_enter)()               */
 146         NULL,                           /* (*bus_exit)()                */
 147         NULL,                           /* (*bus_power)()               */
 148         pcmcia_intr_ops                 /* (*bus_intr_op)();            */
 149 };
______unchanged_portion_omitted_
```

```
**********************************************************
   19077 Wed Feb 26 14:23:24 2014
new/usr/src/uts/common/io/scsi/impl/scsi_resource.c
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
/*
 * 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.
 */

#include <sys/scsi/scsi.h>
#include <sys/vtrace.h>


#define A_TO_TRAN(ap)   ((ap)->a_hba_tran)
#define P_TO_TRAN(pkt)  ((pkt)->pkt_address.a_hba_tran)
#define P_TO_ADDR(pkt)  (&((pkt)->pkt_address))

/*
 * Callback id
 */
uintptr_t scsi_callback_id = 0;

extern ddi_dma_attr_t scsi_alloc_attr;

struct buf *
scsi_alloc_consistent_buf(struct scsi_address *ap,
    struct buf *in_bp, size_t datalen, uint_t bflags,
    int (*callback)(caddr_t), caddr_t callback_arg)
{
        dev_info_t      *pdip;
        struct          buf *bp;
        int             kmflag;
        size_t          rlen;

        TRACE_0(TR_FAC_SCSI_RES, TR_SCSI_ALLOC_CONSISTENT_BUF_START,
            "scsi_alloc_consistent_buf_start");

        if (!in_bp) {
                kmflag = (callback == SLEEP_FUNC) ? KM_SLEEP : KM_NOSLEEP;
                if ((bp = getrbuf(kmflag)) == NULL) {
                        goto no_resource;
                }
        } else {
```



```
                bp = in_bp;

                /* we are establishing a new buffer memory association */
                bp->b_flags &= ~(B_PAGEIO | B_PHYS | B_REMAPPED | B_SHADOW);
                bp->b_proc = NULL;
                bp->b_pages = NULL;
                bp->b_shadow = NULL;
        }

        /* limit bits that can be set by bflags argument */
        ASSERT(!(bflags & ~(B_READ | B_WRITE)));
        bflags &= (B_READ | B_WRITE);
        bp->b_un.b_addr = 0;

        if (datalen) {
                pdip = (A_TO_TRAN(ap))->tran_hba_dip;

                /*
                 * use i_ddi_mem_alloc() for now until we have an interface to
                 * allocate memory for DMA which doesn't require a DMA handle.
                 * ddi_iopb_alloc() is obsolete and we want more flexibility in
                 * controlling the DMA address constraints.
                 */
                while (i_ddi_mem_alloc(pdip, &scsi_alloc_attr, datalen,
                    ((callback == SLEEP_FUNC) ? 1 : 0), 0, NULL,
                    &bp->b_un.b_addr, &rlen, NULL) != DDI_SUCCESS) {
                        if (callback == SLEEP_FUNC) {
                                delay(drv_usectohz(10000));
                        } else {
                                if (!in_bp)
                                        freerbuf(bp);
                                goto no_resource;
                        }
                }
                bp->b_flags |= bflags;
        }
        bp->b_bcount = datalen;
        bp->b_resid = 0;

        TRACE_0(TR_FAC_SCSI_RES, TR_SCSI_ALLOC_CONSISTENT_BUF_END,
            "scsi_alloc_consistent_buf_end");
        return (bp);

no_resource:

        if (callback != NULL_FUNC && callback != SLEEP_FUNC) {
                ddi_set_callback(callback, callback_arg,
                    &scsi_callback_id);
        }
        TRACE_0(TR_FAC_SCSI_RES,
            TR_SCSI_ALLOC_CONSISTENT_BUF_RETURN1_END,
            "scsi_alloc_consistent_buf_end (return1)");
        return (NULL);
}
______unchanged_portion_omitted_
```

****
79551 Wed Feb 26 14:23:24 2014
new/usr/src/uts/common/io/scsi/impl/scsi_subr.c
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
****
_____***unchanged_portion_omitted_***

```
 325 /*
 326  * Common iopbmap data area packet allocation routines
 327  */

 329 struct scsi_pkt *
 330 get_pktiopb(struct scsi_address *ap, caddr_t *datap, int cdblen, int statuslen,
 331     int datalen, int readflag, int (*func)())
 332 {
 333         scsi_hba_tran_t *tran = A_TO_TRAN(ap);
 334         dev_info_t      *pdip = tran->tran_hba_dip;
 335         struct scsi_pkt *pkt = NULL;
 336         struct buf      local;
 337         size_t          rlen;

 339         if (!datap)
 340                 return (pkt);
 341         *datap = (caddr_t)0;
 342         bzero((caddr_t)&local, sizeof (struct buf));

 344         /*
 345          * use i_ddi_mem_alloc() for now until we have an interface to allocate
 346          * memory for DMA which doesn't require a DMA handle.
 346          * memory for DMA which doesn't require a DMA handle. ddi_iopb_alloc()
 347          * is obsolete and we want more flexibility in controlling the DMA
 348          * address constraints.
 347          */
 348         if (i_ddi_mem_alloc(pdip, &scsi_alloc_attr, datalen,
 349             ((func == SLEEP_FUNC) ? 1 : 0), 0, NULL, &local.b_un.b_addr, &rlen,
 350             NULL) != DDI_SUCCESS) {
 351                 return (pkt);
 352         }
 353         if (readflag)
 354                 local.b_flags = B_READ;
 355         local.b_bcount = datalen;
 356         pkt = (*tran->tran_init_pkt) (ap, NULL, &local,
 357             cdblen, statuslen, 0, PKT_CONSISTENT,
 358             (func == SLEEP_FUNC) ? SLEEP_FUNC : NULL_FUNC, NULL);
 359         if (!pkt) {
 360                 i_ddi_mem_free(local.b_un.b_addr, NULL);
 361                 if (func != NULL_FUNC) {
 362                         ddi_set_callback(func, NULL, &scsi_callback_id);
 363                 }
 364         } else {
 365                 *datap = local.b_un.b_addr;
 366         }
 367         return (pkt);
 368 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
  476405 Wed Feb 26 14:23:24 2014
new/usr/src/uts/common/io/scsi/targets/st.c
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
______unchanged_portion_omitted_

1600 static int
1601 st_doattach(struct scsi_device *devp, int (*canwait)())
1602 {
1603         struct scsi_tape *un = NULL;
1604         recov_info *ri;
1605         int km_flags = (canwait != NULL_FUNC) ? KM_SLEEP : KM_NOSLEEP;
1606         int instance;
1607         size_t rlen;

1609         ST_FUNC(devp->sd_dev, st_doattach);
1610         /*
1611          * Call the routine scsi_probe to do some of the dirty work.
1612          * If the INQUIRY command succeeds, the field sd_inq in the
1613          * device structure will be filled in.
1614          */
1615         ST_DEBUG(devp->sd_dev, st_label, SCSI_DEBUG,
1616             "st_doattach(): probing\n");

1618         if (scsi_probe(devp, canwait) == SCSIPROBE_EXISTS) {

1620                 /*
1621                  * In checking the whole inq_dtype byte we are looking at both
1622                  * the Peripheral Qualifier and the Peripheral Device Type.
1623                  * For this driver we are only interested in sequential devices
1624                  * that are connected or capable if connecting to this logical
1625                  * unit.
1626                  */
1627                 if (devp->sd_inq->inq_dtype ==
1628                     (DTYPE_SEQUENTIAL | DPQ_POSSIBLE)) {
1629                         ST_DEBUG(devp->sd_dev, st_label, SCSI_DEBUG,
1630                             "probe exists\n");
1631                 } else {
1632                         /* Something there but not a tape device */
1633                         scsi_unprobe(devp);
1634                         return (DDI_FAILURE);
1635                 }
1636         } else {
1637                 /* Nothing there */
1638                 ST_DEBUG(devp->sd_dev, st_label, SCSI_DEBUG,
1639                     "probe failure: nothing there\n");
1640                 scsi_unprobe(devp);
1641                 return (DDI_FAILURE);
1642         }


1645         /*
1646          * The actual unit is present.
1647          * Now is the time to fill in the rest of our info..
1648          */
1649         instance = ddi_get_instance(devp->sd_dev);

1651         if (ddi_soft_state_zalloc(st_state, instance) != DDI_SUCCESS) {
1652                 goto error;
1653         }
1654         un = ddi_get_soft_state(st_state, instance);

1656         ASSERT(un != NULL);
```



```
1658         un->un_rqs_bp = scsi_alloc_consistent_buf(&devp->sd_address, NULL,
1659             MAX_SENSE_LENGTH, B_READ, canwait, NULL);
1660         if (un->un_rqs_bp == NULL) {
1661                 goto error;
1662         }
1663         un->un_rqs = scsi_init_pkt(&devp->sd_address, NULL, un->un_rqs_bp,
1664             CDB_GROUP0, 1, st_recov_sz, PKT_CONSISTENT, canwait, NULL);
1665         if (!un->un_rqs) {
1666                 goto error;
1667         }
1668         ASSERT(un->un_rqs->pkt_resid == 0);
1669         devp->sd_sense =
1670             (struct scsi_extended_sense *)un->un_rqs_bp->b_un.b_addr;
1671         ASSERT(geterror(un->un_rqs_bp) == NULL);

1673         (void) scsi_setup_cdb((union scsi_cdb *)un->un_rqs->pkt_cdbp,
1674             SCMD_REQUEST_SENSE, 0, MAX_SENSE_LENGTH, 0);
1675         FILL_SCSI1_LUN(devp, un->un_rqs);
1676         un->un_rqs->pkt_flags |= (FLAG_SENSING | FLAG_HEAD | FLAG_NODISCON);
1677         un->un_rqs->pkt_time = st_io_time;
1678         un->un_rqs->pkt_comp = st_intr;
1679         ri = (recov_info *)un->un_rqs->pkt_private;
1680         if (st_recov_sz == sizeof (recov_info)) {
1681                 ri->privatelen = sizeof (recov_info);
1682         } else {
1683                 ri->privatelen = sizeof (pkt_info);
1684         }

1686         un->un_sbufp = getrbuf(km_flags);
1687         un->un_recov_buf = getrbuf(km_flags);

1689         un->un_uscsi_rqs_buf = kmem_alloc(SENSE_LENGTH, KM_SLEEP);

1691         /*
1692          * use i_ddi_mem_alloc() for now until we have an interface to allocate
1693          * memory for DMA which doesn't require a DMA handle.
1693          * memory for DMA which doesn't require a DMA handle. ddi_iopb_alloc()
1694          * is obsolete and we want more flexibility in controlling the DMA
1695          * address constraints.
1694          */
1695         (void) i_ddi_mem_alloc(devp->sd_dev, &st_alloc_attr,
1696             sizeof (struct seq_mode), ((km_flags == KM_SLEEP) ? 1 : 0), 0,
1697             NULL, (caddr_t *)&un->un_mspl, &rlen, NULL);

1699         (void) i_ddi_mem_alloc(devp->sd_dev, &st_alloc_attr,
1700             sizeof (read_pos_data_t), ((km_flags == KM_SLEEP) ? 1 : 0), 0,
1701             NULL, (caddr_t *)&un->un_read_pos_data, &rlen, NULL);

1703         if (!un->un_sbufp || !un->un_mspl || !un->un_read_pos_data) {
1704                 ST_DEBUG6(devp->sd_dev, st_label, SCSI_DEBUG,
1705                     "probe partial failure: no space\n");
1706                 goto error;
1707         }

1709         bzero(un->un_mspl, sizeof (struct seq_mode));

1711         cv_init(&un->un_sbuf_cv, NULL, CV_DRIVER, NULL);
1712         cv_init(&un->un_queue_cv, NULL, CV_DRIVER, NULL);
1713         cv_init(&un->un_clscv, NULL, CV_DRIVER, NULL);
1714         cv_init(&un->un_state_cv, NULL, CV_DRIVER, NULL);
1715 #ifdef  __x86
1716         cv_init(&un->un_contig_mem_cv, NULL, CV_DRIVER, NULL);
1717 #endif

1719         /* Initialize power managemnet condition variable */
1720         cv_init(&un->un_suspend_cv, NULL, CV_DRIVER, NULL);
```

```
1721         cv_init(&un->un_tape_busy_cv, NULL, CV_DRIVER, NULL);
1722         cv_init(&un->un_recov_buf_cv, NULL, CV_DRIVER, NULL);

1724         un->un_recov_taskq = ddi_taskq_create(devp->sd_dev,
1725             "un_recov_taskq", 1, TASKQ_DEFAULTPRI, km_flags);

1727         ASSERT(un->un_recov_taskq != NULL);

1729         un->un_pos.pmode = invalid;
1730         un->un_sd       = devp;
1731         un->un_swr_token = (opaque_t)NULL;
1732         un->un_comp_page = ST_DEV_DATACOMP_PAGE | ST_DEV_CONFIG_PAGE;
1733         un->un_wormable = st_is_drive_worm;
1734         un->un_media_id_method = st_get_media_identification;
1735         /*
1736          * setting long a initial as it contains logical file info.
1737          * support for long format is mandatory but many drive don't do it.
1738          */
1739         un->un_read_pos_type = LONG_POS;

1741         un->un_suspend_pos.pmode = invalid;

1743         st_add_recovery_info_to_pkt(un, un->un_rqs_bp, un->un_rqs);

1745 #ifdef  __x86
1746         if (ddi_dma_alloc_handle(ST_DEVINFO, &st_contig_mem_dma_attr,
1747             DDI_DMA_SLEEP, NULL, &un->un_contig_mem_hdl) != DDI_SUCCESS) {
1748                 ST_DEBUG6(devp->sd_dev, st_label, SCSI_DEBUG,
1749                     "allocation of contiguous memory dma handle failed!");
1750                 un->un_contig_mem_hdl = NULL;
1751                 goto error;
1752         }
1753 #endif

1755         /*
1756          * Since this driver manages devices with "remote" hardware,
1757          * i.e. the devices themselves have no "reg" properties,
1758          * the SUSPEND/RESUME commands in detach/attach will not be
1759          * called by the power management framework unless we request
1760          * it by creating a "pm-hardware-state" property and setting it
1761          * to value "needs-suspend-resume".
1762          */
1763         if (ddi_prop_update_string(DDI_DEV_T_NONE, devp->sd_dev,
1764             "pm-hardware-state", "needs-suspend-resume") !=
1765             DDI_PROP_SUCCESS) {

1767                 ST_DEBUG(devp->sd_dev, st_label, SCSI_DEBUG,
1768                     "ddi_prop_update(\"pm-hardware-state\") failed\n");
1769                 goto error;
1770         }

1772         if (ddi_prop_create(DDI_DEV_T_NONE, devp->sd_dev, DDI_PROP_CANSLEEP,
1773             "no-involuntary-power-cycles", NULL, 0) != DDI_PROP_SUCCESS) {

1775                 ST_DEBUG(devp->sd_dev, st_label, SCSI_DEBUG,
1776                     "ddi_prop_create(\"no-involuntary-power-cycles\") "
1777                     "failed\n");
1778                 goto error;
1779         }

1781         (void) scsi_reset_notify(ROUTE, SCSI_RESET_NOTIFY,
1782             st_reset_notification, (caddr_t)un);

1784         ST_DEBUG6(devp->sd_dev, st_label, SCSI_DEBUG, "attach success\n");
1785         return (DDI_SUCCESS);
```



```
1787 error:
1788         devp->sd_sense = NULL;

1790         ddi_remove_minor_node(devp->sd_dev, NULL);
1791         if (un) {
1792                 if (un->un_mspl) {
1793                         i_ddi_mem_free((caddr_t)un->un_mspl, NULL);
1794                 }
1795                 if (un->un_read_pos_data) {
1796                         i_ddi_mem_free((caddr_t)un->un_read_pos_data, 0);
1797                 }
1798                 if (un->un_sbufp) {
1799                         freerbuf(un->un_sbufp);
1800                 }
1801                 if (un->un_recov_buf) {
1802                         freerbuf(un->un_recov_buf);
1803                 }
1804                 if (un->un_uscsi_rqs_buf) {
1805                         kmem_free(un->un_uscsi_rqs_buf, SENSE_LENGTH);
1806                 }
1807 #ifdef  __x86
1808                 if (un->un_contig_mem_hdl != NULL) {
1809                         ddi_dma_free_handle(&un->un_contig_mem_hdl);
1810                 }
1811 #endif
1812                 if (un->un_rqs) {
1813                         scsi_destroy_pkt(un->un_rqs);
1814                 }

1816                 if (un->un_rqs_bp) {
1817                         scsi_free_consistent_buf(un->un_rqs_bp);
1818                 }

1820                 ddi_soft_state_free(st_state, instance);
1821                 devp->sd_private = NULL;
1822         }

1824         if (devp->sd_inq) {
1825                 scsi_unprobe(devp);
1826         }
1827         return (DDI_FAILURE);
1828 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    5046 Wed Feb 26 14:23:25 2014
new/usr/src/uts/common/io/warlock/ddi_dki_comm.inc
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
```

```
/*
 * 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 2012 Garrett D'Amore <garrett@damore.org>.
 * Copyright 2014 Pluribus Networks, Inc.
 * Copyright 2012 Garrett D'Amore <garrett@damore.org>.  All rights reserved.
 */

/*
 * ddi_dki_comm.inc - Part of a pseudo-kernel to use when analyzing drivers
 * with warlock.
 *
 * The main idea here is to represent all of the ways that the kernel can
 * call into the driver, so that warlock has the correct view of the call
 * graph.
 *
 * This file represents the stuff in common between the DDI/DKI spec and
 * the current implementation.  It is included by both ddi_dki_{spec,impl}.c
 *
 * This is a SPARC version; some functions (e.g. ddi_dma_nextwin) should
 * be changed for an x86 version.
 */

#include <sys/note.h>
#include <sys/devops.h>
#include <sys/ddi.h>
#include <sys/sunddi.h>
#include <sys/proc.h>

_NOTE(DATA_READABLE_WITHOUT_LOCK( dev_ops cb_ops bus_ops ))

/*
 * Now define a dev_ops, a cb_ops, and a bus_ops with 0 for each
 * entry point, so that warlock doesn't complain that these
 * function pointers have no bindings.
 *      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
 */
```



```
struct dev_ops *devops_p, warlock_dev_ops = {
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
};
```

_____***unchanged_portion_omitted_***

```
int
ddi_dma_addr_setup(
        dev_info_t *a,
        struct as *b,
        caddr_t c,
        size_t d,
        uint_t e,
        int (*fp)(void),
        caddr_t f,
        ddi_dma_lim_t *g,
        ddi_dma_handle_t *h)
{
        struct bus_ops *ops;
        (*ops->bus_dma_map)(0, 0, 0, 0);
        thread_create(0, 0, (void (*)())fp, 0, 0, 0, 0, 0);
}

int
ddi_dma_buf_setup(
        dev_info_t *a,
        struct buf *b,
        uint_t c,
        int (*fp)(void),
        caddr_t d,
        ddi_dma_lim_t *e,
        ddi_dma_handle_t *f)
{
        struct bus_ops *ops;
        (*ops->bus_dma_map)(0, 0, 0, 0);
        thread_create(0, 0, (void (*)())fp, 0, 0, 0, 0, 0);
}
```

```
void
ddi_set_callback(
        int (*fp)(caddr_t),
        caddr_t a,
        uintptr_t *b)
{
        thread_create(0, 0, (void (*)())fp, 0, 0, 0, 0, 0);
}
```

_____***unchanged_portion_omitted_***

```
int
ddi_dma_setup(
        dev_info_t *a,
        struct ddi_dma_req *b,
        ddi_dma_handle_t *c)
{
        struct bus_ops *ops;
        (*ops->bus_dma_map)(0, 0, 0, 0);
}

int
ddi_dma_mctl(dev_info_t *a, dev_info_t *b, ddi_dma_handle_t c,
        enum ddi_dma_ctlops d, off_t *e, size_t *f, caddr_t *g,
        uint_t h)
{
        struct bus_ops *ops;
        (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
}
```

```
160 int
202 ddi_dma_kvaddrp(ddi_dma_handle_t h, off_t off, size_t len, caddr_t *kp)
203 {
204         struct bus_ops *ops;
205         (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
206 }

208 int
209 ddi_dma_htoc(ddi_dma_handle_t h, off_t o, ddi_dma_cookie_t *c)
210 {
211         struct bus_ops *ops;
212         (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
213 }

215 int
216 ddi_dma_coff(ddi_dma_handle_t h, ddi_dma_cookie_t *c, off_t *o)
217 {
218         struct bus_ops *ops;
219         (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
220 }

222 int
223 ddi_dma_get_error(ddi_dma_handle_t h, uint_t len, caddr_t errblk)
224 {
225         struct bus_ops *ops;
226         (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
227 }

229 int
230 ddi_dma_segtocookie(ddi_dma_seg_t seg, off_t *o, off_t *l,
231     ddi_dma_cookie_t *cookiep)
232 {
233         struct bus_ops *ops;
234         (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
235 }

237 int
161 ddi_dma_sync(ddi_dma_handle_t h, off_t o, size_t l, uint_t whom)
162 {
163         struct bus_ops *ops;
241         (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
242 }

244 int
245 ddi_dma_free(ddi_dma_handle_t h)
246 {
247         struct bus_ops *ops;
248         (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
249 }

251 int
252 ddi_iopb_alloc(dev_info_t *dip, ddi_dma_lim_t *limp, uint_t len, caddr_t *iopbp)
253 {
254         struct bus_ops *ops;
255         (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
256 }

258 void
259 ddi_iopb_free(caddr_t iopb)
260 {
261         struct bus_ops *ops;
262         (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
263 }

265 int
266 ddi_mem_alloc(dev_info_t *dip, ddi_dma_lim_t *limits, uint_t length,
```



```
267         uint_t flags, caddr_t *kaddrp, uint_t *real_length)
268 {
269         struct bus_ops *ops;
270         (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
271 }

273 void
274 ddi_mem_free(caddr_t kaddr)
275 {
276         struct bus_ops *ops;
164         (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
165 }
```

_____unchanged_portion_omitted_

```
**********************************************************
   25529 Wed Feb 26 14:23:25 2014
new/usr/src/uts/common/sys/ddidmareq.h
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
______unchanged_portion_omitted_

 505 /*
 506  * Defines for the DMA mapping allocation functions
 507  *
 508  * If a DMA callback funtion is set to anything other than the following
 509  * defines then it is assumed that one wishes a callback and is providing
 510  * a function address.
 511  */
 512 #ifdef __STDC__
 513 #define DDI_DMA_DONTWAIT        ((int (*)(caddr_t))0)
 514 #define DDI_DMA_SLEEP           ((int (*)(caddr_t))1)
 515 #else
 516 #define DDI_DMA_DONTWAIT        ((int (*)())0)
 517 #define DDI_DMA_SLEEP           ((int (*)())1)
 518 #endif

 520 /*
 521  * Return values from callback functions.
 522  */
 523 #define DDI_DMA_CALLBACK_RUNOUT 0
 524 #define DDI_DMA_CALLBACK_DONE   1

 526 /*
 527  * Flag definitions for the allocation functions.
 528  */
 529 #define DDI_DMA_WRITE           0x0001  /* Direction memory --> IO      */
 530 #define DDI_DMA_READ            0x0002  /* Direction IO --> memory      */
 531 #define DDI_DMA_RDWR            (DDI_DMA_READ | DDI_DMA_WRITE)

 533 /*
 534  * If possible, establish a MMU redzone after the mapping (to protect
 535  * against cheap DMA hardware that might get out of control).
 536  */
 537 #define DDI_DMA_REDZONE         0x0004

 539 /*
 540  * A partial allocation is allowed. That is, if the size of the object
 541  * exceeds the mapping resources available, only map a portion of the
 542  * object and return status indicating that this took place. The caller
 543  * can use the functions ddi_dma_numwin(9F) and ddi_dma_getwin(9F) to
 544  * change, at a later point, the actual mapped portion of the object.
 545  *
 546  * The mapped portion begins at offset 0 of the object.
 547  *
 548  */
 549 #define DDI_DMA_PARTIAL         0x0008

 551 /*
 552  * Map the object for byte consistent access. Note that explicit
 553  * synchronization (via ddi_dma_sync(9F)) will still be required.
 554  * Consider this flag to be a hint to the mapping routines as to
 555  * the intended use of the mapping.
 556  *
 557  * Normal data transfers can be usually consider to use 'streaming'
 558  * modes of operations. They start at a specific point, transfer a
 559  * fairly large amount of data sequentially, and then stop (usually
 560  * on a well aligned boundary).
 561  *
 562  * Control mode data transfers (for memory resident device control blocks,
```



```
 563  * e.g., ethernet message descriptors) do not access memory in such
 564  * a streaming sequential fashion. Instead, they tend to modify a few
 565  * words or bytes, move around and maybe modify a few more.
 566  *
 567  * There are many machine implementations that make this difficult to
 568  * control in a generic and seamless fashion. Therefore, explicit synch-
 569  * ronization steps (via ddi_dma_sync(9F)) are still required (even if you
 570  * ask for a byte-consistent mapping) in order to make the view of the
 571  * memory object shared between a CPU and a DMA master in consistent.
 572  * However, judicious use of this flag can give sufficient hints to
 573  * the mapping routines to attempt to pick the most efficacious mapping
 574  * such that the synchronization steps are as efficient as possible.
 575  *
 576  */
 577 #define DDI_DMA_CONSISTENT      0x0010

 579 /*
 580  * Some DMA mappings have to be 'exclusive' access.
 581  */
 582 #define DDI_DMA_EXCLUSIVE       0x0020

 584 /*
 585  * Sequential, unidirectional, block-sized and block aligned transfers
 586  */
 587 #define DDI_DMA_STREAMING       0x0040

 589 /*
 590  * Support for 64-bit SBus devices
 591  */
 592 #define DDI_DMA_SBUS_64BIT      0x2000

 594 /*
 595  * Return values from the mapping allocation functions.
 596  */

 598 /*
 599  * succeeded in satisfying request
 600  */
 601 #define DDI_DMA_MAPPED          0

 603 /*
 604  * Mapping is legitimate (for advisory calls).
 605  */
 606 #define DDI_DMA_MAPOK           0

 608 /*
 609  * Succeeded in mapping a portion of the request.
 610  */
 611 #define DDI_DMA_PARTIAL_MAP     1

 613 /*
 614  * indicates end of window/segment list
 615  */
 616 #define DDI_DMA_DONE            2

 618 /*
 619  * No resources to map request.
 620  */
 621 #define DDI_DMA_NORESOURCES     -1

 623 /*
 624  * Can't establish a mapping to the specified object
 625  * (no specific reason).
 626  */
 627 #define DDI_DMA_NOMAPPING       -2
```

```
 629 /*
 630  * The request is too big to be mapped.
 631  */
 632 #define DDI_DMA_TOOBIG          -3

 634 /*
 635  * The request is too small to be mapped.
 636  */
 637 #define DDI_DMA_TOOSMALL        -4

 639 /*
 640  * The request cannot be mapped because the object
 641  * is locked against mapping by another DMA master.
 642  */
 643 #define DDI_DMA_LOCKED          -5

 645 /*
 646  * The request cannot be mapped because the limits
 647  * structure has bogus values.
 648  */
 649 #define DDI_DMA_BADLIMITS       -6

 651 /*
 652  * the segment/window pointer is stale
 653  */
 654 #define DDI_DMA_STALE           -7

 656 /*
 657  * The system can't allocate DMA resources using
 658  * the given DMA attributes
 659  */
 660 #define DDI_DMA_BADATTR         -8

 662 /*
 663  * A DMA handle is already used for a DMA
 664  */
 665 #define DDI_DMA_INUSE           -9


 668 /*
 669  * DVMA disabled or not supported. use physical DMA
 670  */
 671 #define DDI_DMA_USE_PHYSICAL            -10


 674 /*
 675  * In order for the access to a memory object to be consistent
 676  * between a device and a CPU, the function ddi_dma_sync(9F)
 677  * must be called upon the DMA handle. The following flags
 678  * define whose view of the object should be made consistent.
 679  * There are different flags here because on different machines
 680  * there are definite performance implications of how long
 681  * such synchronization takes.
 682  *
 683  * DDI_DMA_SYNC_FORDEV makes all device references to the object
 684  * mapped by the DMA handle up to date. It should be used by a
 685  * driver after a cpu modifies the memory object (over the range
 686  * specified by the other arguments to the ddi_dma_sync(9F) call).
 687  *
 688  * DDI_DMA_SYNC_FORCPU makes all cpu references to the object
 689  * mapped by the DMA handle up to date. It should be used
 690  * by a driver after the receipt of data from the device to
 691  * the memory object is done (over the range specified by
 692  * the other arguments to the ddi_dma_sync(9F) call).
 693  *
 694  * If the only mapping that concerns the driver is one for the kernel,
```



```
 695  * the flag DDI_DMA_SYNC_FORKERNEL can be used. This is a hint to the
 694  * If the only mapping that concerns the driver is one for the
 695  * kernel (such as memory allocated by ddi_iopb_alloc(9F)), the
 696  * flag DDI_DMA_SYNC_FORKERNEL can be used. This is a hint to the
 696  * system that if it can synchronize the kernel's view faster
 697  * that the CPU's view, it can do so, otherwise it acts the
 698  * same as DDI_DMA_SYNC_FORCPU. DDI_DMA_SYNC_FORKERNEL might
 699  * speed up the synchronization of kernel mappings in case of
 700  * non IO-coherent CPU caches.
 701  */
 702 #define DDI_DMA_SYNC_FORDEV     0x0
 703 #define DDI_DMA_SYNC_FORCPU     0x1
 704 #define DDI_DMA_SYNC_FORKERNEL  0x2

 706 /*
 707  * Bus nexus control functions for DMA
 708  */

 710 /*
 711  * Control operations, defined here so that devops.h can be included
 712  * by drivers without having to include a specific SYSDDI implementation
 713  * header file.
 714  */

 716 enum ddi_dma_ctlops {
 717         DDI_DMA_FREE,           /* obsolete - do not use                */
 718         DDI_DMA_SYNC,           /* obsolete - do not use                */
 719         DDI_DMA_HTOC,           /* obsolete - do not use                */
 720         DDI_DMA_KVADDR,         /* obsolete - do not use                */
 721         DDI_DMA_MOVWIN,         /* obsolete - do not use                */
 722         DDI_DMA_REPWIN,         /* obsolete - do not use                */
 723         DDI_DMA_GETERR,         /* obsolete - do not use                */
 724         DDI_DMA_COFF,           /* obsolete - do not use                */
 725         DDI_DMA_NEXTWIN,        /* obsolete - do not use                */
 726         DDI_DMA_NEXTSEG,        /* obsolete - do not use                */
 727         DDI_DMA_SEGTOC,         /* obsolete - do not use                */
 728         DDI_DMA_RESERVE,        /* reserve some DVMA range              */
 729         DDI_DMA_RELEASE,        /* free preallocated DVMA range         */
 730         DDI_DMA_RESETH,         /* obsolete - do not use                */
 731         DDI_DMA_CKSYNC,         /* obsolete - do not use                */
 732         DDI_DMA_IOPB_ALLOC,     /* get contiguous DMA-able memory       */
 733         DDI_DMA_IOPB_FREE,      /* return contiguous DMA-able memory    */
 734         DDI_DMA_SMEM_ALLOC,     /* get contiguous DMA-able memory       */
 735         DDI_DMA_SMEM_FREE,      /* return contiguous DMA-able memory    */
 736         DDI_DMA_SET_SBUS64,     /* 64 bit SBus support                  */
 737         DDI_DMA_REMAP,          /* remap DMA buffers after relocation   */

 739                 /*
 740                  * control ops for DMA engine on motherboard
 741                  */
 742         DDI_DMA_E_ACQUIRE,      /* get channel for exclusive use        */
 743         DDI_DMA_E_FREE,         /* release channel                      */
 744         DDI_DMA_E_1STPTY,       /* setup channel for 1st party DMA      */
 745         DDI_DMA_E_GETCB,        /* get control block for DMA engine     */
 746         DDI_DMA_E_FREECB,       /* free control blk for DMA engine      */
 747         DDI_DMA_E_PROG,         /* program channel of DMA engine        */
 748         DDI_DMA_E_SWSETUP,      /* setup channel for software control   */
 749         DDI_DMA_E_SWSTART,      /* software operation of DMA channel    */
 750         DDI_DMA_E_ENABLE,       /* enable channel of DMA engine         */
 751         DDI_DMA_E_STOP,         /* stop a channel of DMA engine         */
 752         DDI_DMA_E_DISABLE,      /* disable channel of DMA engine        */
 753         DDI_DMA_E_GETCNT,       /* get remaining xfer count             */
 754         DDI_DMA_E_GETLIM,       /* get DMA engine limits                */
 755         DDI_DMA_E_GETATTR       /* get DMA engine attributes            */
 756 };
_____unchanged_portion_omitted_
```

```
**********************************************************
    9805 Wed Feb 26 14:23:25 2014
new/usr/src/uts/common/sys/scsi/conf/device.h
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
/*
 * 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.
 */

/*
 * SCSI device structure.
 *
 * All SCSI target drivers will have one of these per target/lun/sfunc.
 * It is allocated and initialized by the framework SCSA HBA nexus code
 * for each SCSI target dev_info_t node during HBA nexus DDI_CTLOPS_INITCHILD
 * processing of a child device node just prior to tran_tgt_init(9E).  A
 * pointer the the scsi_device(9S) structure is stored in the
 * driver-private data field of the target device's dev_info_t node (in
 * 'devi_driver_data') and can be retrieved by ddi_get_driver_private(9F).
 */
#ifndef _SYS_SCSI_CONF_DEVICE_H
#define _SYS_SCSI_CONF_DEVICE_H

#include <sys/scsi/scsi_types.h>

#ifdef  __cplusplus
extern "C" {
#endif

struct scsi_device {
        /*
         * Routing information for a SCSI device (target/lun/sfunc).
         *
         * The scsi_address(9S) structure contains a pointer to the
         * scsi_hba_tran(9S) of the transport.
         *
         * For devices below an HBA that uses SCSI_HBA_ADDR_SPI
         * unit-addressing, the scsi_address(9S) information contains
         * decoded target/lun addressing information.
         *
         * For devices below an HBA that uses SCSI_HBA_ADDR_COMPLEX
         * unit-addressing, the scsi_address(9S) information contains a
         * pointer to the scsi_device(9S) structure and the HBA can maintain
         * its private per-unit-address/per-scsi_device information using
```



```
         * scsi_address_device(9F) and scsi_device_hba_private_[gs]et(9F).
         *
         * NOTE: The scsi_address(9S) structure gets structure-copied into
         * the scsi_pkt(9S) 'pkt_address' field. Having a pointer to the
         * scsi_device(9S) structure within the scsi_address(9S) allows
         * the SCSA framework to reflect generic changes in device state
         * at scsi_pkt_comp(9F) time (given just a scsi_pkt(9S) pointer).
         *
         * NOTE: The older SCSI_HBA_TRAN_CLONE method of supporting
         * SCSI-3 devices is still supported, but use is discouraged.
         */
        struct scsi_address     sd_address;

        /* Cross-reference to target device's dev_info_t. */
        dev_info_t              *sd_dev;

        /*
         * Target driver mutex for this device. Initialized by SCSA HBA
         * framework code prior to probe(9E) or attach(9E) of scsi_device.
         */
        kmutex_t                sd_mutex;

        /*
         * SCSA private: use is associated with implementation of
         * SCSI_HBA_ADDR_COMPLEX scsi_device_hba_private_[gs]et(9F).
         * The HBA driver can store a pointer to per-scsi_device(9S)
         * HBA private data during its tran_tgt_init(9E) implementation
         * by calling scsi_device_hba_private_set(9F), and free that
         * pointer during tran_tgt_fini(9E). At tran_send(9E) time, the
         * HBA driver can use scsi_address_device(9F) to obtain a pointer
         * to the scsi_device(9S) structure, and then gain access to
         * its per-scsi_device(9S) hba private data by calling
         * scsi_device_hba_private_get(9F).
         */
        void                    *sd_hba_private;

        /*
         * If scsi_slave is used to probe out this device, a scsi_inquiry data
         * structure will be allocated and an INQUIRY command will be run to
         * fill it in.
         *
         * The allocation will be done via ddi_iopb_alloc, so any manual
         * freeing may be done by ddi_iopb_free.
         *
         * The inquiry data is allocated/refreshed by scsi_probe/scsi_slave
         * and freed by uninitchild (inquiry data is no longer freed by
         * scsi_unprobe/scsi_unslave).
         *
         * NOTE: Additional device identity information may be available
         * as properties of sd_dev.
         */
        struct scsi_inquiry     *sd_inq;

        /*
         * Place to point to an extended request sense buffer.
         * The target driver is responsible for managing this.
         */
        struct scsi_extended_sense      *sd_sense;

        /*
         * Target driver 'private' information. Typically a pointer to target
         * driver private ddi_soft_state(9F) information for the device.  This
         * information is typically established in target driver attach(9E),
         * and freed in the target driver detach(9E).
         *
         * LEGACY: For a scsi_device structure allocated by scsi_vhci during
```

```
124          * online of a path, this was set by scsi_vhci to point to the
125          * pathinfo node. Please use sd_pathinfo instead.
126          */
127         void                    *sd_private;

129         /*
130          * FMA capabilities of scsi_device.
131          */
132         int                     sd_fm_capable;

134         /*
135          * mdi_pathinfo_t pointer to pathinfo node for scsi_device structure
136          * allocated by the scsi_vhci for transport to a specific pHCI path.
137          */
138         void                    *sd_pathinfo;

140         /*
141          * sd_uninit_prevent - Counter that prevents demotion of
142          * DS_INITIALIZED node (esp loss of devi_addr) by causing
143          * DDI_CTLOPS_UNINITCHILD failure - devi_ref will not protect
144          * demotion of DS_INITIALIZED node.
145          *
146          * sd_tran_tgt_free_done - in some cases SCSA will call
147          * tran_tgt_free(9E) independent of devinfo node state, this means
148          * that uninitchild code should not call tran_tgt_free(9E).
149          */
150         int                     sd_uninit_prevent:16,
151                                 sd_tran_tgt_free_done:1,
152                                 sd_flags_pad:15;

154         /*
155          * The 'sd_tran_safe' field is a grotty hack that allows direct-access
156          * (non-scsa) drivers (like chs, ata, and mlx - which all make cmdk
157          * children) to *illegally* put their own vector in the scsi_address(9S)
158          * 'a_hba_tran' field. When all the drivers that overwrite
159          * 'a_hba_tran' are fixed, we can remove sd_tran_safe (and make
160          * scsi_hba.c code trust that the 'sd_address.a_hba_tran' established
161          * during initchild is still valid when uninitchild occurs).
162          *
163          * NOTE: This hack is also shows up in the DEVP_TO_TRAN implementation
164          * in scsi_confsubr.c.
165          *
166          * NOTE: The 'sd_tran_safe' field is only referenced by SCSA framework
167          * code, so always keeping it at the end of the scsi_device structure
168          * (until it can be removed) is OK.  It use to be called 'sd_reserved'.
169          */
170         struct scsi_hba_tran    *sd_tran_safe;

172 #ifdef  SCSI_SIZE_CLEAN_VERIFY
173         /*
174          * Must be last: Building a driver with-and-without
175          * -DSCSI_SIZE_CLEAN_VERIFY, and checking driver modules for
176          * differences with a tools like 'wsdiff' allows a developer to verify
177          * that their driver has no dependencies on scsi*(9S) size.
178          */
179         int                     _pad[8];
180 #endif  /* SCSI_SIZE_CLEAN_VERIFY */
181 };
```

*unchanged_portion_omitted_*

```
**********************************************************
   62830 Wed Feb 26 14:23:25 2014
new/usr/src/uts/common/sys/sunddi.h
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
______unchanged_portion_omitted_
 485 #endif  /* _MEMCHR_INLINE */
 486 #else
 487 extern void *memchr(const void *, int, size_t);
 488 #endif /* __cplusplus >= 199711L */

 490 extern int ddi_strtol(const char *, char **, int, long *);
 491 extern int ddi_strtoul(const char *, char **, int, unsigned long *);
 492 extern int ddi_strtoll(const char *, char **, int, longlong_t *);
 493 extern int ddi_strtoull(const char *, char **, int, u_longlong_t *);

 495 /*
 496  * kiconv functions and their macros.
 497  */
 498 #define KICONV_IGNORE_NULL      (0x0001)
 499 #define KICONV_REPLACE_INVALID  (0x0002)

 501 extern kiconv_t kiconv_open(const char *, const char *);
 502 extern size_t kiconv(kiconv_t, char **, size_t *, char **, size_t *, int *);
 503 extern int kiconv_close(kiconv_t);
 504 extern size_t kiconvstr(const char *, const char *, char *, size_t *, char *,
 505         size_t *, int, int *);

 507 /*
 508  * ddi_map_regs
 509  *
 510  *      Map in the register set given by rnumber.
 511  *      The register number determine which register
 512  *      set will be mapped if more than one exists.
 513  *      The parent driver gets the information
 514  *      from parent private data and sets up the
 515  *      appropriate mappings and returns the kernel
 516  *      virtual address of the register set in *kaddrp.
 517  *      The offset specifies an offset into the register
 518  *      space to start from and len indicates the size
 519  *      of the area to map. If len and offset are 0 then
 520  *      the entire space is mapped.  It returns DDI_SUCCESS on
 521  *      success or DDI_FAILURE otherwise.
 522  *
 523  */
 524 int
 525 ddi_map_regs(dev_info_t *dip, uint_t rnumber, caddr_t *kaddrp,
 526         off_t offset, off_t len);

 528 /*
 529  * ddi_unmap_regs
 530  *
 531  *      Undo mappings set up by ddi_map_regs.
 532  *      The register number determines which register
 533  *      set will be unmapped if more than one exists.
 534  *      This is provided for drivers preparing
 535  *      to detach themselves from the system to
 536  *      allow them to release allocated mappings.
 537  *
 538  *      The kaddrp and len specify the area to be
 539  *      unmapped. *kaddrp was returned from ddi_map_regs
 540  *      and len should match what ddi_map_regs was called
 541  *      with.
 542  */
```



```
 544 void
 545 ddi_unmap_regs(dev_info_t *dip, uint_t rnumber, caddr_t *kaddrp,
 546         off_t offset, off_t len);

 548 int
 549 ddi_map(dev_info_t *dp, ddi_map_req_t *mp, off_t offset, off_t len,
 550         caddr_t *addrp);

 552 int
 553 ddi_apply_range(dev_info_t *dip, dev_info_t *rdip, struct regspec *rp);

 555 /*
 556  * ddi_rnumber_to_regspec: Not for use by leaf drivers.
 557  */
 558 struct regspec *
 559 ddi_rnumber_to_regspec(dev_info_t *dip, int rnumber);

 561 int
 562 ddi_bus_map(dev_info_t *dip, dev_info_t *rdip, ddi_map_req_t *mp, off_t offset,
 563         off_t len, caddr_t *vaddrp);

 565 int
 566 nullbusmap(dev_info_t *dip, dev_info_t *rdip, ddi_map_req_t *mp, off_t offset,
 567         off_t len, caddr_t *vaddrp);

 569 int ddi_peek8(dev_info_t *dip, int8_t *addr, int8_t *val_p);
 570 int ddi_peek16(dev_info_t *dip, int16_t *addr, int16_t *val_p);
 571 int ddi_peek32(dev_info_t *dip, int32_t *addr, int32_t *val_p);
 572 int ddi_peek64(dev_info_t *dip, int64_t *addr, int64_t *val_p);

 574 int ddi_poke8(dev_info_t *dip, int8_t *addr, int8_t val);
 575 int ddi_poke16(dev_info_t *dip, int16_t *addr, int16_t val);
 576 int ddi_poke32(dev_info_t *dip, int32_t *addr, int32_t val);
 577 int ddi_poke64(dev_info_t *dip, int64_t *addr, int64_t val);

 579 /*
 580  * Peek and poke to and from a uio structure in xfersize pieces,
 581  * using the parent nexi.
 582  */
 583 int ddi_peekpokeio(dev_info_t *devi, struct uio *uio, enum uio_rw rw,
 584         caddr_t addr, size_t len, uint_t xfersize);

 586 /*
 587  * Pagesize conversions using the parent nexi
 588  */
 589 unsigned long ddi_btop(dev_info_t *dip, unsigned long bytes);
 590 unsigned long ddi_btopr(dev_info_t *dip, unsigned long bytes);
 591 unsigned long ddi_ptob(dev_info_t *dip, unsigned long pages);

 593 /*
 594  * There are no more "block" interrupt functions, per se.
 595  * All thread of control should be done with MP/MT lockings.
 596  *
 597  * However, there are certain times in which a driver needs
 598  * absolutely a critical guaranteed non-preemptable time
 599  * in which to execute a few instructions.
 600  *
 601  * The following pair of functions attempt to guarantee this,
 602  * but they are dangerous to use. That is, use them with
 603  * extreme care. They do not guarantee to stop other processors
 604  * from executing, but they do guarantee that the caller
 605  * of ddi_enter_critical will continue to run until the
 606  * caller calls ddi_exit_critical. No intervening DDI functions
 607  * may be called between an entry and an exit from a critical
 608  * region.
 609  *
```

```
 610  * ddi_enter_critical returns an integer identifier which must
 611  * be passed to ddi_exit_critical.
 612  *
 613  * Be very sparing in the use of these functions since it is
 614  * likely that absolutely nothing else can occur in the system
 615  * whilst in the critical region.
 616  */

 618 unsigned int
 619 ddi_enter_critical(void);

 621 void
 622 ddi_exit_critical(unsigned int);

 624 /*
 625  * devmap functions
 626  */
 627 int
 628 devmap_setup(dev_t dev, offset_t off, ddi_as_handle_t as, caddr_t *addrp,
 629         size_t len, uint_t prot, uint_t maxprot, uint_t flags,
 630         struct cred *cred);

 632 int
 633 ddi_devmap_segmap(dev_t dev, off_t off, ddi_as_handle_t as, caddr_t *addrp,
 634         off_t len, uint_t prot, uint_t maxprot, uint_t flags,
 635         struct cred *cred);

 637 int
 638 devmap_load(devmap_cookie_t dhp, offset_t offset, size_t len, uint_t type,
 639         uint_t rw);

 641 int
 642 devmap_unload(devmap_cookie_t dhp, offset_t offset, size_t len);

 644 int
 645 devmap_devmem_setup(devmap_cookie_t dhp, dev_info_t *dip,
 646         struct devmap_callback_ctl *callback_ops,
 647         uint_t rnumber, offset_t roff, size_t len, uint_t maxprot,
 648         uint_t flags, ddi_device_acc_attr_t *accattrp);

 650 int
 651 devmap_umem_setup(devmap_cookie_t dhp, dev_info_t *dip,
 652         struct devmap_callback_ctl *callback_ops,
 653         ddi_umem_cookie_t cookie, offset_t off, size_t len, uint_t maxprot,
 654         uint_t flags, ddi_device_acc_attr_t *accattrp);

 656 int
 657 devmap_devmem_remap(devmap_cookie_t dhp, dev_info_t *dip,
 658         uint_t rnumber, offset_t roff, size_t len, uint_t maxprot,
 659         uint_t flags, ddi_device_acc_attr_t *accattrp);

 661 int
 662 devmap_umem_remap(devmap_cookie_t dhp, dev_info_t *dip,
 663         ddi_umem_cookie_t cookie, offset_t off, size_t len, uint_t maxprot,
 664         uint_t flags, ddi_device_acc_attr_t *accattrp);

 666 void
 667 devmap_set_ctx_timeout(devmap_cookie_t dhp, clock_t ticks);

 669 int
 670 devmap_default_access(devmap_cookie_t dhp, void *pvtp, offset_t off,
 671         size_t len, uint_t type, uint_t rw);

 673 int
 674 devmap_do_ctxmgt(devmap_cookie_t dhp, void *pvtp, offset_t off, size_t len,
 675         uint_t type, uint_t rw, int (*ctxmgt)(devmap_cookie_t, void *, offset_t,
```



```
 676         size_t, uint_t, uint_t));


 679 void *ddi_umem_alloc(size_t size, int flag, ddi_umem_cookie_t *cookiep);

 681 void ddi_umem_free(ddi_umem_cookie_t cookie);

 683 /*
 684  * Functions to lock user memory and do repeated I/O or do devmap_umem_setup
 685  */
 686 int
 687 ddi_umem_lock(caddr_t addr, size_t size, int flags, ddi_umem_cookie_t *cookie);

 689 void
 690 ddi_umem_unlock(ddi_umem_cookie_t cookie);

 692 struct buf *
 693 ddi_umem_iosetup(ddi_umem_cookie_t cookie, off_t off, size_t len, int direction,
 694     dev_t dev, daddr_t blkno, int (*iodone)(struct buf *), int sleepflag);

 696 /*
 697  * Mapping functions
 698  */
 699 int
 700 ddi_segmap(dev_t dev, off_t offset, struct as *asp, caddr_t *addrp, off_t len,
 701         uint_t prot, uint_t maxprot, uint_t flags, cred_t *credp);

 703 int
 704 ddi_segmap_setup(dev_t dev, off_t offset, struct as *as, caddr_t *addrp,
 705         off_t len, uint_t prot, uint_t maxprot, uint_t flags, cred_t *cred,
 706         ddi_device_acc_attr_t *accattrp, uint_t rnumber);

 708 int
 709 ddi_map_fault(dev_info_t *dip, struct hat *hat, struct seg *seg, caddr_t addr,
 710         struct devpage *dp, pfn_t pfn, uint_t prot, uint_t lock);

 712 int
 713 ddi_device_mapping_check(dev_t dev, ddi_device_acc_attr_t *accattrp,
 714         uint_t rnumber, uint_t *hat_flags);

 716 /*
 717  * Property functions:   See also, ddipropdefs.h.
 718  *                       In general, the underlying driver MUST be held
 719  *                       to call it's property functions.
 720  */

 722 /*
 723  * Used to create, modify, and lookup integer properties
 724  */
 725 int ddi_prop_get_int(dev_t match_dev, dev_info_t *dip, uint_t flags,
 726     char *name, int defvalue);
 727 int64_t ddi_prop_get_int64(dev_t match_dev, dev_info_t *dip, uint_t flags,
 728     char *name, int64_t defvalue);
 729 int ddi_prop_lookup_int_array(dev_t match_dev, dev_info_t *dip, uint_t flags,
 730     char *name, int **data, uint_t *nelements);
 731 int ddi_prop_lookup_int64_array(dev_t match_dev, dev_info_t *dip, uint_t flags,
 732     char *name, int64_t **data, uint_t *nelements);
 733 int ddi_prop_update_int(dev_t match_dev, dev_info_t *dip,
 734     char *name, int data);
 735 int ddi_prop_update_int64(dev_t match_dev, dev_info_t *dip,
 736     char *name, int64_t data);
 737 int ddi_prop_update_int_array(dev_t match_dev, dev_info_t *dip,
 738     char *name, int *data, uint_t nelements);
 739 int ddi_prop_update_int64_array(dev_t match_dev, dev_info_t *dip,
 740     char *name, int64_t *data, uint_t nelements);
 741 /*
```

```
 742  * Used to create, modify, and lookup string properties
 743  */
 744 int ddi_prop_lookup_string(dev_t match_dev, dev_info_t *dip, uint_t flags,
 745     char *name, char **data);
 746 int ddi_prop_lookup_string_array(dev_t match_dev, dev_info_t *dip, uint_t flags,
 747     char *name, char ***data, uint_t *nelements);
 748 int ddi_prop_update_string(dev_t match_dev, dev_info_t *dip,
 749     char *name, char *data);
 750 int ddi_prop_update_string_array(dev_t match_dev, dev_info_t *dip,
 751     char *name, char **data, uint_t nelements);

 753 /*
 754  * Used to create, modify, and lookup byte properties
 755  */
 756 int ddi_prop_lookup_byte_array(dev_t match_dev, dev_info_t *dip, uint_t flags,
 757     char *name, uchar_t **data, uint_t *nelements);
 758 int ddi_prop_update_byte_array(dev_t match_dev, dev_info_t *dip,
 759     char *name, uchar_t *data, uint_t nelements);

 761 /*
 762  * Used to verify the existence of a property or to see if a boolean
 763  * property exists.
 764  */
 765 int ddi_prop_exists(dev_t match_dev, dev_info_t *dip, uint_t flags, char *name);

 767 /*
 768  * Used to free the data returned by the above property routines.
 769  */
 770 void ddi_prop_free(void *data);

 772 /*
 773  * nopropop: For internal use in 'dummy' cb_prop_op functions only
 774  */

 776 int
 777 nopropop(dev_t dev, dev_info_t *dip, ddi_prop_op_t prop_op, int mod_flags,
 778         char *name, caddr_t valuep, int *lengthp);

 780 /*
 781  * ddi_prop_op: The basic property operator for drivers.
 782  *
 783  * In ddi_prop_op, the type of valuep is interpreted based on prop_op:
 784  *
 785  *      prop_op                 valuep
 786  *      ------                  ------
 787  *
 788  *      PROP_LEN                <unused>
 789  *
 790  *      PROP_LEN_AND_VAL_BUF    Pointer to callers buffer
 791  *
 792  *      PROP_LEN_AND_VAL_ALLOC  Address of callers pointer (will be set to
 793  *                              address of allocated buffer, if successful)
 794  */

 796 int
 797 ddi_prop_op(dev_t dev, dev_info_t *dip, ddi_prop_op_t prop_op, int mod_flags,
 798         char *name, caddr_t valuep, int *lengthp);

 800 /* ddi_prop_op_size: for drivers that implement size in bytes */
 801 int
 802 ddi_prop_op_size(dev_t dev, dev_info_t *dip, ddi_prop_op_t prop_op,
 803         int mod_flags, char *name, caddr_t valuep, int *lengthp,
 804         uint64_t size64);

 806 /* ddi_prop_op_size_blksize: like ddi_prop_op_size, in blksize blocks */
 807 int
```



```
 808 ddi_prop_op_size_blksize(dev_t dev, dev_info_t *dip, ddi_prop_op_t prop_op,
 809         int mod_flags, char *name, caddr_t valuep, int *lengthp,
 810         uint64_t size64, uint_t blksize);

 812 /* ddi_prop_op_nblocks: for drivers that implement size in DEV_BSIZE blocks */
 813 int
 814 ddi_prop_op_nblocks(dev_t dev, dev_info_t *dip, ddi_prop_op_t prop_op,
 815         int mod_flags, char *name, caddr_t valuep, int *lengthp,
 816         uint64_t nblocks64);

 818 /* ddi_prop_op_nblocks_blksize: like ddi_prop_op_nblocks, in blksize blocks */
 819 int
 820 ddi_prop_op_nblocks_blksize(dev_t dev, dev_info_t *dip, ddi_prop_op_t prop_op,
 821         int mod_flags, char *name, caddr_t valuep, int *lengthp,
 822         uint64_t nblocks64, uint_t blksize);

 824 /*
 825  * Variable length props...
 826  */

 828 /*
 829  * ddi_getlongprop:     Get variable length property len+val into a buffer
 830  *              allocated by property provider via kmem_alloc. Requester
 831  *              is responsible for freeing returned property via kmem_free.
 832  *
 833  *      Arguments:
 834  *
 835  *      dev:    Input:  dev_t of property.
 836  *      dip:    Input:  dev_info_t pointer of child.
 837  *      flags:  Input:  Possible flag modifiers are:
 838  *              DDI_PROP_DONTPASS:      Don't pass to parent if prop not found.
 839  *              DDI_PROP_CANSLEEP:      Memory allocation may sleep.
 840  *      name:   Input:  name of property.
 841  *      valuep: Output: Addr of callers buffer pointer.
 842  *      lengthp:Output: *lengthp will contain prop length on exit.
 843  *
 844  *      Possible Returns:
 845  *
 846  *              DDI_PROP_SUCCESS:       Prop found and returned.
 847  *              DDI_PROP_NOT_FOUND:     Prop not found
 848  *              DDI_PROP_UNDEFINED:     Prop explicitly undefined.
 849  *              DDI_PROP_NO_MEMORY:     Prop found, but unable to alloc mem.
 850  */

 852 int
 853 ddi_getlongprop(dev_t dev, dev_info_t *dip, int flags,
 854         char *name, caddr_t valuep, int *lengthp);

 856 /*
 857  *
 858  * ddi_getlongprop_buf:         Get long prop into pre-allocated callers
 859  *                              buffer. (no memory allocation by provider).
 860  *
 861  *      dev:    Input:  dev_t of property.
 862  *      dip:    Input:  dev_info_t pointer of child.
 863  *      flags:  Input:  DDI_PROP_DONTPASS or NULL
 864  *      name:   Input:  name of property
 865  *      valuep: Input:  ptr to callers buffer.
 866  *      lengthp:I/O:    ptr to length of callers buffer on entry,
 867  *                      actual length of property on exit.
 868  *
 869  *      Possible returns:
 870  *
 871  *              DDI_PROP_SUCCESS        Prop found and returned
 872  *              DDI_PROP_NOT_FOUND      Prop not found
 873  *              DDI_PROP_UNDEFINED      Prop explicitly undefined.
```

```
 *              DDI_PROP_BUF_TOO_SMALL  Prop found, callers buf too small,
 *                                      no value returned, but actual prop
 *                                      length returned in *lengthp
 *
 */

int
ddi_getlongprop_buf(dev_t dev, dev_info_t *dip, int flags,
        char *name, caddr_t valuep, int *lengthp);

/*
 * Integer/boolean sized props.
 *
 * Call is value only... returns found boolean or int sized prop value or
 * defvalue if prop not found or is wrong length or is explicitly undefined.
 * Only flag is DDI_PROP_DONTPASS...
 *
 * By convention, this interface returns boolean (0) sized properties
 * as value (int)1.
 */

int
ddi_getprop(dev_t dev, dev_info_t *dip, int flags, char *name, int defvalue);

/*
 * Get prop length interface: flags are 0 or DDI_PROP_DONTPASS
 * if returns DDI_PROP_SUCCESS, length returned in *lengthp.
 */

int
ddi_getproplen(dev_t dev, dev_info_t *dip, int flags, char *name, int *lengthp);


/*
 * Interface to create/modify a managed property on child's behalf...
 * Only flag is DDI_PROP_CANSLEEP to allow memory allocation to sleep
 * if no memory available for internal prop structure.  Long property
 * (non integer sized) value references are not copied.
 *
 * Define property with DDI_DEV_T_NONE dev_t for properties not associated
 * with any particular dev_t. Use the same dev_t when modifying or undefining
 * a property.
 *
 * No guarantee on order of property search, so don't mix the same
 * property name with wildcard and non-wildcard dev_t's.
 */

/*
 * ddi_prop_create:     Define a managed property:
 */

int
ddi_prop_create(dev_t dev, dev_info_t *dip, int flag,
        char *name, caddr_t value, int length);

/*
 * ddi_prop_modify:     Modify a managed property value
 */

int
ddi_prop_modify(dev_t dev, dev_info_t *dip, int flag,
        char *name, caddr_t value, int length);

/*
 * ddi_prop_remove:     Undefine a managed property:
 */
```



```
int
ddi_prop_remove(dev_t dev, dev_info_t *dip, char *name);

/*
 * ddi_prop_remove_all:         Used before unloading a driver to remove
 *                              all properties. (undefines all dev_t's props.)
 *                              Also removes 'undefined' prop defs.
 */

void
ddi_prop_remove_all(dev_info_t *dip);


/*
 * ddi_prop_undefine:   Explicitly undefine a property.  Property
 *                      searches which match this property return
 *                      the error code DDI_PROP_UNDEFINED.
 *
 *                      Use ddi_prop_remove to negate effect of
 *                      ddi_prop_undefine
 */

int
ddi_prop_undefine(dev_t dev, dev_info_t *dip, int flag, char *name);


/*
 * ddi_prop_cache_invalidate
 *                      Invalidate a property in the current cached
 *                      devinfo snapshot - next cached snapshot will
 *                      return the latest property value available.
 */
void
ddi_prop_cache_invalidate(dev_t dev, dev_info_t *dip, char *name, int flags);

/*
 * The default ddi_bus_prop_op wrapper...
 */

int
ddi_bus_prop_op(dev_t dev, dev_info_t *dip, dev_info_t *ch_dip,
        ddi_prop_op_t prop_op, int mod_flags,
        char *name, caddr_t valuep, int *lengthp);


/*
 * Routines to traverse the tree of dev_info nodes.
 * The general idea of these functions is to provide
 * various tree traversal utilities. For each node
 * that the tree traversal function finds, a caller
 * supplied function is called with arguments of
 * the current node and a caller supplied argument.
 * The caller supplied function should return one
 * of the integer values defined below which will
 * indicate to the tree traversal function whether
 * the traversal should be continued, and if so, how,
 * or whether the traversal should terminate.
 */

/*
 * This general-purpose routine traverses the tree of dev_info nodes,
 * starting from the given node, and calls the given function for each
 * node that it finds with the current node and the pointer arg (which
 * can point to a structure of information that the function
 * needs) as arguments.
```

```
1006  *
1007  * It does the walk a layer at a time, not depth-first.
1008  *
1009  * The given function must return one of the values defined above.
1010  *
1011  */

1013 void
1014 ddi_walk_devs(dev_info_t *, int (*)(dev_info_t *, void *), void *);

1016 /*
1017  * Routines to get at elements of the dev_info structure
1018  */

1020 /*
1021  * ddi_node_name gets the device's 'name' from the device node.
1022  *
1023  * ddi_binding_name gets the string the OS used to bind the node to a driver,
1024  * in certain cases, the binding name may be different from the node name,
1025  * if the node name does not name a specific device driver.
1026  *
1027  * ddi_get_name is a synonym for ddi_binding_name().
1028  */
1029 char *
1030 ddi_get_name(dev_info_t *dip);

1032 char *
1033 ddi_binding_name(dev_info_t *dip);

1035 const char *
1036 ddi_driver_name(dev_info_t *dip);

1038 major_t
1039 ddi_driver_major(dev_info_t *dip);

1041 major_t
1042 ddi_compatible_driver_major(dev_info_t *dip, char **formp);

1044 char *
1045 ddi_node_name(dev_info_t *dip);

1047 int
1048 ddi_get_nodeid(dev_info_t *dip);

1050 int
1051 ddi_get_instance(dev_info_t *dip);

1053 struct dev_ops *
1054 ddi_get_driver(dev_info_t *dip);

1056 void
1057 ddi_set_driver(dev_info_t *dip, struct dev_ops *devo);

1059 void
1060 ddi_set_driver_private(dev_info_t *dip, void *data);

1062 void *
1063 ddi_get_driver_private(dev_info_t *dip);

1065 /*
1066  * ddi_dev_is_needed tells system that a device is about to use a
1067  * component. Returns when component is ready.
1068  */
1069 int
1070 ddi_dev_is_needed(dev_info_t *dip, int cmpt, int level);
```



```
1072 /*
1073  * check if DDI_SUSPEND may result in power being removed from a device.
1074  */
1075 int
1076 ddi_removing_power(dev_info_t *dip);

1078 /*
1079  *  (Obsolete) power entry point
1080  */
1081 int
1082 ddi_power(dev_info_t *dip, int cmpt, int level);

1084 /*
1085  * ddi_get_parent requires that the branch of the tree with the
1086  * node be held (ddi_hold_installed_driver) or that the devinfo tree
1087  * lock be held
1088  */
1089 dev_info_t *
1090 ddi_get_parent(dev_info_t *dip);

1092 /*
1093  * ddi_get_child and ddi_get_next_sibling require that the devinfo
1094  * tree lock be held
1095  */
1096 dev_info_t *
1097 ddi_get_child(dev_info_t *dip);

1099 dev_info_t *
1100 ddi_get_next_sibling(dev_info_t *dip);

1102 dev_info_t *
1103 ddi_get_next(dev_info_t *dip);

1105 void
1106 ddi_set_next(dev_info_t *dip, dev_info_t *nextdip);

1108 /*
1109  * dev_info manipulation functions
1110  */

1112 /*
1113  * Add and remove child devices. These are part of the system framework.
1114  *
1115  * ddi_add_child creates a dev_info structure with the passed name,
1116  * nodeid and instance arguments and makes it a child of pdip. Devices
1117  * that are known directly by the hardware have real nodeids; devices
1118  * that are software constructs use the defined DEVI_PSEUDO_NODEID
1119  * for the node id.
1120  *
1121  * ddi_remove_node removes the node from the tree. This fails if this
1122  * child has children. Parent and driver private data should already
1123  * be released (freed) prior to calling this function.  If flag is
1124  * non-zero, the child is removed from it's linked list of instances.
1125  */
1126 dev_info_t *
1127 ddi_add_child(dev_info_t *pdip, char *name, uint_t nodeid, uint_t instance);

1129 int
1130 ddi_remove_child(dev_info_t *dip, int flag);

1132 /*
1133  * Given the major number for a driver, make sure that dev_info nodes
1134  * are created form the driver's hwconf file, the driver for the named
1135  * device is loaded and attached, as well as any drivers for parent devices.
1136  * Return a pointer to the driver's dev_ops struct with the dev_ops held.
1137  * Note - Callers must release the dev_ops with ddi_rele_driver.
```

```
1138  *
1139  * When a driver is held, the branch of the devinfo tree from any of the
1140  * drivers devinfos to the root node are automatically held.  This only
1141  * applies to tree traversals up (and back down) the tree following the
1142  * parent pointers.
1143  *
1144  * Use of this interface is discouraged, it may be removed in a future release.
1145  */
1146 struct dev_ops *
1147 ddi_hold_installed_driver(major_t major);

1149 void
1150 ddi_rele_driver(major_t major);

1152 /*
1153  * Attach and hold the specified instance of a driver.  The flags argument
1154  * should be zero.
1155  */
1156 dev_info_t *
1157 ddi_hold_devi_by_instance(major_t major, int instance, int flags);

1159 void
1160 ddi_release_devi(dev_info_t *);

1162 /*
1163  * Associate a streams queue with a devinfo node
1164  */
1165 void
1166 ddi_assoc_queue_with_devi(queue_t *, dev_info_t *);

1168 /*
1169  * Given the identifier string passed, make sure that dev_info nodes
1170  * are created form the driver's hwconf file, the driver for the named
1171  * device is loaded and attached, as well as any drivers for parent devices.
1172  *
1173  * Note that the driver is not held and is subject to being removed the instant
1174  * this call completes.  You probably really want ddi_hold_installed_driver.
1175  */
1176 int
1177 ddi_install_driver(char *idstring);

1179 /*
1180  * Routines that return specific nodes
1181  */

1183 dev_info_t *
1184 ddi_root_node(void);

1186 /*
1187  * Given a name and an instance number, find and return the
1188  * dev_info from the current state of the device tree.
1189  *
1190  * If instance number is -1, return the first named instance.
1191  *
1192  * If attached is 1, exclude all nodes that are < DS_ATTACHED
1193  *
1194  * Requires that the devinfo tree be locked.
1195  * If attached is 1, the driver must be held.
1196  */
1197 dev_info_t *
1198 ddi_find_devinfo(char *name, int instance, int attached);

1200 /*
1201  * Synchronization of I/O with respect to various
1202  * caches and system write buffers.
1203  *
```



```
1204  * Done at varying points during an I/O transfer (including at the
1205  * removal of an I/O mapping).
1206  *
1207  * Due to the support of systems with write buffers which may
1208  * not be able to be turned off, this function *must* used at
1209  * any point in which data consistency might be required.
1210  *
1211  * Generally this means that if a memory object has multiple mappings
1212  * (both for I/O, as described by the handle, and the IU, via, e.g.
1213  * a call to ddi_dma_kvaddrp), and one mapping may have been
1214  * used to modify the memory object, this function must be called
1215  * to ensure that the modification of the memory object is
1216  * complete, as well as possibly to inform other mappings of
1217  * the object that any cached references to the object are
1218  * now stale (and flush or invalidate these stale cache references
1219  * as necessary).
1220  *
1221  * The function ddi_dma_sync() provides the general interface with
1222  * respect to this capability. Generally, ddi_dma_unbind_handle()
1223  * (below) may be used in preference to ddi_dma_sync() as it calls
1222  * respect to this capability. Generally, ddi_dma_free() (below) may
1223  * be used in preference to ddi_dma_sync() as ddi_dma_free() calls
1224  * ddi_dma_sync().
1225  *
1226  * Returns 0 if all caches that exist and are specified by cache_flags
1227  * are successfully operated on, else -1.
1228  *
1229  * The argument offset specifies an offset into the mapping of the mapped
1230  * object in which to perform the synchronization. It will be silently
1231  * truncated to the granularity of underlying cache line sizes as
1232  * appropriate.
1233  *
1234  * The argument len specifies a length starting from offset in which to
1235  * perform the synchronization. A value of (uint_t) -1 means that the length
1236  * proceeds from offset to the end of the mapping. The length argument
1237  * will silently rounded up to the granularity of underlying cache line
1238  * sizes  as appropriate.
1239  *
1240  * The argument flags specifies what to synchronize (the device's view of
1241  * the object or the cpu's view of the object).
1242  *
1243  * Inquiring minds want to know when ddi_dma_sync should be used:
1244  *
1245  * +    When an object is mapped for dma, assume that an
1246  *      implicit ddi_dma_sync() is done for you.
1247  *
1248  * +    When an object is unmapped (ddi_dma_unbind_handle()), assume
1248  * +    When an object is unmapped (ddi_dma_free()), assume
1249  *      that an implicit ddi_dma_sync() is done for you.
1250  *
1251  * +    At any time between the two times above that the
1252  *      memory object may have been modified by either
1253  *      the DMA device or a processor and you wish that
1254  *      the change be noticed by the master that didn't
1255  *      do the modifying.
1256  *
1257  * Clearly, only the third case above requires the use of ddi_dma_sync.
1258  *
1259  * Inquiring minds also want to know which flag to use:
1260  *
1261  * +    If you *modify* with a cpu the object, you use
1262  *      ddi_dma_sync(...DDI_DMA_SYNC_FORDEV) (you are making sure
1263  *      that the DMA device sees the changes you made).
1264  *
1265  * +    If you are checking, with the processor, an area
1266  *      of the object that the DMA device *may* have modified,
```

```
1267  *      you use ddi_dma_sync(....DDI_DMA_SYNC_FORCPU) (you are
1268  *      making sure that the processor(s) will see the changes
1269  *      that the DMA device may have made).
1270  */

1272 int
1273 ddi_dma_sync(ddi_dma_handle_t handle, off_t offset, size_t len, uint_t flags);

1275 /*
1276  * Return the allowable DMA burst size for the object mapped by handle.
1277  * The burst sizes will returned in an integer that encodes power
1278  * of two burst sizes that are allowed in bit encoded format. For
1279  * example, a transfer that could allow 1, 2, 4, 8 and 32 byte bursts
1280  * would be encoded as 0x2f. A transfer that could be allowed as solely
1281  * a halfword (2 byte) transfers would be returned as 0x2.
1282  */

1284 int
1285 ddi_dma_burstsizes(ddi_dma_handle_t handle);

1287 /*
1288  * Merge DMA attributes
1289  */

1291 void
1292 ddi_dma_attr_merge(ddi_dma_attr_t *attr, ddi_dma_attr_t *mod);

1294 /*
1295  * Allocate a DMA handle
1296  */

1298 int
1299 ddi_dma_alloc_handle(dev_info_t *dip, ddi_dma_attr_t *attr,
1300         int (*waitfp)(caddr_t), caddr_t arg,
1301         ddi_dma_handle_t *handlep);

1303 /*
1304  * Free DMA handle
1305  */

1307 void
1308 ddi_dma_free_handle(ddi_dma_handle_t *handlep);

1310 /*
1311  * Allocate memory for DMA transfers
1312  */

1314 int
1315 ddi_dma_mem_alloc(ddi_dma_handle_t handle, size_t length,
1316         ddi_device_acc_attr_t *accattrp, uint_t xfermodes,
1317         int (*waitfp)(caddr_t), caddr_t arg, caddr_t *kaddrp,
1318         size_t *real_length, ddi_acc_handle_t *handlep);

1320 /*
1321  * Free DMA memory
1322  */

1324 void
1325 ddi_dma_mem_free(ddi_acc_handle_t *hp);

1327 /*
1328  * bind address to a DMA handle
1329  */

1331 int
1332 ddi_dma_addr_bind_handle(ddi_dma_handle_t handle, struct as *as,
```



```
1333         caddr_t addr, size_t len, uint_t flags,
1334         int (*waitfp)(caddr_t), caddr_t arg,
1335         ddi_dma_cookie_t *cookiep, uint_t *ccountp);

1337 /*
1338  * bind buffer to DMA handle
1339  */

1341 int
1342 ddi_dma_buf_bind_handle(ddi_dma_handle_t handle, struct buf *bp,
1343         uint_t flags, int (*waitfp)(caddr_t), caddr_t arg,
1344         ddi_dma_cookie_t *cookiep, uint_t *ccountp);

1346 /*
1347  * unbind mapping object to handle
1348  */

1350 int
1351 ddi_dma_unbind_handle(ddi_dma_handle_t handle);

1353 /*
1354  * get next DMA cookie
1355  */

1357 void
1358 ddi_dma_nextcookie(ddi_dma_handle_t handle, ddi_dma_cookie_t *cookiep);

1360 /*
1361  * get number of DMA windows
1362  */

1364 int
1365 ddi_dma_numwin(ddi_dma_handle_t handle, uint_t *nwinp);

1367 /*
1368  * get specific DMA window
1369  */

1371 int
1372 ddi_dma_getwin(ddi_dma_handle_t handle, uint_t win, off_t *offp,
1373         size_t *lenp, ddi_dma_cookie_t *cookiep, uint_t *ccountp);

1375 /*
1376  * activate 64 bit SBus support
1377  */

1379 int
1380 ddi_dma_set_sbus64(ddi_dma_handle_t handle, ulong_t burstsizes);

1382 /*
1383  * Miscellaneous functions
1384  */

1386 /*
1387  * ddi_report_dev:      Report a successful attach.
1388  */

1390 void
1391 ddi_report_dev(dev_info_t *dev);

1393 /*
1394  * ddi_dev_regsize
1395  *
1396  *      If the device has h/w register(s), report
1397  *      the size, in bytes, of the specified one into *resultp.
1398  *
```

```
1399  *      Returns DDI_FAILURE if there are not registers,
1400  *      or the specified register doesn't exist.
1401  */

1403 int
1404 ddi_dev_regsize(dev_info_t *dev, uint_t rnumber, off_t *resultp);

1406 /*
1407  * ddi_dev_nregs
1408  *
1409  *      If the device has h/w register(s), report
1410  *      how many of them that there are into resultp.
1411  *      Return DDI_FAILURE if the device has no registers.
1412  */

1414 int
1415 ddi_dev_nregs(dev_info_t *dev, int *resultp);

1417 /*
1418  * ddi_dev_is_sid
1419  *
1420  *      If the device is self-identifying, i.e.,
1421  *      has already been probed by a smart PROM
1422  *      (and thus registers are known to be valid)
1423  *      return DDI_SUCCESS, else DDI_FAILURE.
1424  */


1427 int
1428 ddi_dev_is_sid(dev_info_t *dev);

1430 /*
1431  * ddi_slaveonly
1432  *
1433  *      If the device is on a bus that precludes
1434  *      the device from being either a dma master or
1435  *      a dma slave, return DDI_SUCCESS.
1436  */

1438 int
1439 ddi_slaveonly(dev_info_t *);


1442 /*
1443  * ddi_dev_affinity
1444  *
1445  *      Report, via DDI_SUCCESS, whether there exists
1446  *      an 'affinity' between two dev_info_t's. An
1447  *      affinity is defined to be either a parent-child,
1448  *      or a sibling relationship such that the siblings
1449  *      or in the same part of the bus they happen to be
1450  *      on.
1451  */

1453 int
1454 ddi_dev_affinity(dev_info_t *deva, dev_info_t *devb);


1457 /*
1458  * ddi_set_callback
1459  *
1460  *      Set a function/arg pair into the callback list identified
1461  *      by listid. *listid must always initially start out as zero.
1462  */

1464 void
```



```
1465 ddi_set_callback(int (*funcp)(caddr_t), caddr_t arg, uintptr_t *listid);

1467 /*
1468  * ddi_run_callback
1469  *
1470  *      Run the callback list identified by listid.
1471  */

1473 void
1474 ddi_run_callback(uintptr_t *listid);

1476 /*
1477  * More miscellaneous
1478  */

1480 int
1481 nochpoll(dev_t dev, short events, int anyyet, short *reventsp,
1482         struct pollhead **phpp);

1484 dev_info_t *
1485 nodevinfo(dev_t dev, int otyp);

1487 int
1488 ddi_no_info(dev_info_t *dip, ddi_info_cmd_t infocmd, void *arg, void **result);

1490 int
1491 ddi_getinfo_1to1(dev_info_t *dip, ddi_info_cmd_t infocmd,
1492     void *arg, void **result);

1494 int
1495 ddifail(dev_info_t *devi, ddi_attach_cmd_t cmd);

1497 int
1498 ddi_no_dma_map(dev_info_t *dip, dev_info_t *rdip,
1499     struct ddi_dma_req *dmareqp, ddi_dma_handle_t *handlep);

1501 int
1502 ddi_no_dma_allochdl(dev_info_t *dip, dev_info_t *rdip, ddi_dma_attr_t *attr,
1503     int (*waitfp)(caddr_t), caddr_t arg, ddi_dma_handle_t *handlep);

1505 int
1506 ddi_no_dma_freehdl(dev_info_t *dip, dev_info_t *rdip,
1507     ddi_dma_handle_t handle);

1509 int
1510 ddi_no_dma_bindhdl(dev_info_t *dip, dev_info_t *rdip,
1511     ddi_dma_handle_t handle, struct ddi_dma_req *dmareq,
1512     ddi_dma_cookie_t *cp, uint_t *ccountp);

1514 int
1515 ddi_no_dma_unbindhdl(dev_info_t *dip, dev_info_t *rdip,
1516     ddi_dma_handle_t handle);

1518 int
1519 ddi_no_dma_flush(dev_info_t *dip, dev_info_t *rdip,
1520     ddi_dma_handle_t handle, off_t off, size_t len,
1521     uint_t cache_flags);

1523 int
1524 ddi_no_dma_win(dev_info_t *dip, dev_info_t *rdip,
1525     ddi_dma_handle_t handle, uint_t win, off_t *offp,
1526     size_t *lenp, ddi_dma_cookie_t *cookiep, uint_t *ccountp);

1528 int
1529 ddi_no_dma_mctl(register dev_info_t *dip, dev_info_t *rdip,
1530     ddi_dma_handle_t handle, enum ddi_dma_ctlops request,
```

```
1531     off_t *offp, size_t *lenp, caddr_t *objp, uint_t flags);

1533 void
1534 ddivoid();

1536 cred_t *
1537 ddi_get_cred(void);

1539 time_t
1540 ddi_get_time(void);

1542 pid_t
1543 ddi_get_pid(void);

1545 kt_did_t
1546 ddi_get_kt_did(void);

1548 boolean_t
1549 ddi_can_receive_sig(void);

1551 void
1552 swab(void *src, void *dst, size_t nbytes);

1554 int
1555 ddi_create_minor_node(dev_info_t *dip, char *name, int spec_type,
1556     minor_t minor_num, char *node_type, int flag);

1558 int
1559 ddi_create_priv_minor_node(dev_info_t *dip, char *name, int spec_type,
1560     minor_t minor_num, char *node_type, int flag,
1561     const char *rdpriv, const char *wrpriv, mode_t priv_mode);

1563 void
1564 ddi_remove_minor_node(dev_info_t *dip, char *name);

1566 int
1567 ddi_in_panic(void);

1569 int
1570 ddi_streams_driver(dev_info_t *dip);

1572 /*
1573  * DDI wrappers for ffs and fls
1574  */
1575 int
1576 ddi_ffs(long mask);

1578 int
1579 ddi_fls(long mask);

1581 /*
1582  * The ddi_soft_state* routines comprise generic storage management utilities
1583  * for driver soft state structures.  Two types of soft_state indexes are
1584  * supported: 'integer index', and 'string index'.
1585  */
1586 typedef struct __ddi_soft_state_bystr   ddi_soft_state_bystr;

1588 /*
1589  * Initialize a soft_state set, establishing the 'size' of soft state objects
1590  * in the set.
1591  *
1592  * For an 'integer indexed' soft_state set, the initial set will accommodate
1593  * 'n_items' objects - 'n_items' is a hint (i.e. zero is allowed), allocations
1594  * that exceed 'n_items' have additional overhead.
1595  *
1596  * For a 'string indexed' soft_state set, 'n_items' should be the typical
```



```
1597  * number of soft state objects in the set - 'n_items' is a hint, there may
1598  * be additional overhead if the hint is too small (and wasted memory if the
1599  * hint is too big).
1600  */
1601 int
1602 ddi_soft_state_init(void **state_p, size_t size, size_t n_items);
1603 int
1604 ddi_soft_state_bystr_init(ddi_soft_state_bystr **state_p,
1605     size_t size, int n_items);

1607 /*
1608  * Allocate a soft state object associated with either 'integer index' or
1609  * 'string index' from a soft_state set.
1610  */
1611 int
1612 ddi_soft_state_zalloc(void *state, int item);
1613 int
1614 ddi_soft_state_bystr_zalloc(ddi_soft_state_bystr *state, const char *str);

1616 /*
1617  * Get the pointer to the allocated soft state object associated with
1618  * either 'integer index' or 'string index'.
1619  */
1620 void *
1621 ddi_get_soft_state(void *state, int item);
1622 void *
1623 ddi_soft_state_bystr_get(ddi_soft_state_bystr *state, const char *str);

1625 /*
1626  * Free the soft state object associated with either 'integer index'
1627  * or 'string index'.
1628  */
1629 void
1630 ddi_soft_state_free(void *state, int item);
1631 void
1632 ddi_soft_state_bystr_free(ddi_soft_state_bystr *state, const char *str);

1634 /*
1635  * Free the soft state set and any associated soft state objects.
1636  */
1637 void
1638 ddi_soft_state_fini(void **state_p);
1639 void
1640 ddi_soft_state_bystr_fini(ddi_soft_state_bystr **state_p);

1642 /*
1643  * The ddi_strid_* routines provide string-to-index management utilities.
1644  */
1645 typedef struct __ddi_strid      ddi_strid;
1646 int
1647 ddi_strid_init(ddi_strid **strid_p, int n_items);
1648 id_t
1649 ddi_strid_alloc(ddi_strid *strid, char *str);
1650 id_t
1651 ddi_strid_str2id(ddi_strid *strid, char *str);
1652 char *
1653 ddi_strid_id2str(ddi_strid *strid, id_t id);
1654 void
1655 ddi_strid_free(ddi_strid *strid, id_t id);
1656 void
1657 ddi_strid_fini(ddi_strid **strid_p);

1659 /*
1660  * Set the addr field of the name in dip to name
1661  */
1662 void
```

```
1663 ddi_set_name_addr(dev_info_t *dip, char *name);

1665 /*
1666  * Get the address part of the name.
1667  */
1668 char *
1669 ddi_get_name_addr(dev_info_t *dip);

1671 void
1672 ddi_set_parent_data(dev_info_t *dip, void *pd);

1674 void *
1675 ddi_get_parent_data(dev_info_t *dip);

1677 int
1678 ddi_initchild(dev_info_t *parent, dev_info_t *proto);

1680 int
1681 ddi_uninitchild(dev_info_t *dip);

1683 major_t
1684 ddi_name_to_major(char *name);

1686 char *
1687 ddi_major_to_name(major_t major);

1689 char *
1690 ddi_deviname(dev_info_t *dip, char *name);

1692 char *
1693 ddi_pathname(dev_info_t *dip, char *path);

1695 char *
1696 ddi_pathname_minor(struct ddi_minor_data *dmdp, char *path);

1698 char *
1699 ddi_pathname_obp(dev_info_t *dip, char *path);

1701 int
1702 ddi_pathname_obp_set(dev_info_t *dip, char *component);

1704 int
1705 ddi_dev_pathname(dev_t devt, int spec_type, char *name);

1707 dev_t
1708 ddi_pathname_to_dev_t(char *pathname);

1710 /*
1711  * High resolution system timer functions.
1712  *
1713  * These functions are already in the kernel (see sys/time.h).
1714  * The ddi supports the notion of a hrtime_t type and the
1715  * functions gethrtime, hrtadd, hrtsub and hrtcmp.
1716  */


1719 /*
1720  * Nexus wrapper functions
1721  *
1722  * These functions are for entries in a bus nexus driver's bus_ops
1723  * structure for when the driver doesn't have such a function and
1724  * doesn't wish to prohibit such a function from existing. They
1725  * may also be called to start passing a request up the dev_info
1726  * tree.
1727  */
```



```
1729 /*
1730  * bus_ctl wrapper
1731  */

1733 int
1734 ddi_ctlops(dev_info_t *d, dev_info_t *r, ddi_ctl_enum_t o, void *a, void *v);

1736 /*
1737  * bus_dma_map wrapper
1738  */

1740 int
1741 ddi_dma_allochdl(dev_info_t *dip, dev_info_t *rdip, ddi_dma_attr_t *attr,
1742         int (*waitfp)(caddr_t), caddr_t arg, ddi_dma_handle_t *handlep);

1744 int
1745 ddi_dma_freehdl(dev_info_t *dip, dev_info_t *rdip,
1746         ddi_dma_handle_t handle);

1748 int
1749 ddi_dma_bindhdl(dev_info_t *dip, dev_info_t *rdip,
1750         ddi_dma_handle_t handle, struct ddi_dma_req *dmareq,
1751         ddi_dma_cookie_t *cp, uint_t *ccountp);

1753 int
1754 ddi_dma_unbindhdl(dev_info_t *dip, dev_info_t *rdip,
1755         ddi_dma_handle_t handle);

1757 int
1758 ddi_dma_flush(dev_info_t *dip, dev_info_t *rdip,
1759         ddi_dma_handle_t handle, off_t off, size_t len,
1760         uint_t cache_flags);

1762 int
1763 ddi_dma_win(dev_info_t *dip, dev_info_t *rdip,
1764         ddi_dma_handle_t handle, uint_t win, off_t *offp,
1765         size_t *lenp, ddi_dma_cookie_t *cookiep, uint_t *ccountp);

1767 /*
1768  * bus_dma_ctl wrapper
1769  */

1771 int
1772 ddi_dma_mctl(dev_info_t *dip, dev_info_t *rdip, ddi_dma_handle_t handle,
1773         enum ddi_dma_ctlops request, off_t *offp, size_t *lenp,
1774         caddr_t *objp, uint_t flags);

1776 /*
1777  * dvma support for networking drivers
1778  */

1780 unsigned long
1781 dvma_pagesize(dev_info_t *dip);

1783 int
1784 dvma_reserve(dev_info_t *dip,  ddi_dma_lim_t *limp, uint_t pages,
1785         ddi_dma_handle_t *handlep);

1787 void
1788 dvma_release(ddi_dma_handle_t h);

1790 void
1791 dvma_kaddr_load(ddi_dma_handle_t h, caddr_t a, uint_t len, uint_t index,
1792         ddi_dma_cookie_t *cp);

1794 void
```

```
1795 dvma_unload(ddi_dma_handle_t h, uint_t objindex, uint_t type);

1797 void
1798 dvma_sync(ddi_dma_handle_t h, uint_t objindex, uint_t type);

1800 /*
1801  * Layered driver support
1802  */

1804 extern int ddi_copyin(const void *, void *, size_t, int);
1805 extern int ddi_copyout(const void *, void *, size_t, int);

1807 /*
1808  * Send signals to processes
1809  */
1810 extern void *proc_ref(void);
1811 extern void proc_unref(void *pref);
1812 extern int proc_signal(void *pref, int sig);

1814 /* I/O port access routines */
1815 extern uint8_t inb(int port);
1816 extern uint16_t inw(int port);
1817 extern uint32_t inl(int port);
1818 extern void outb(int port, uint8_t value);
1819 extern void outw(int port, uint16_t value);
1820 extern void outl(int port, uint32_t value);

1822 /*
1823  * Console bell routines
1824  */
1825 extern void ddi_ring_console_bell(clock_t duration);
1826 extern void ddi_set_console_bell(void (*bellfunc)(clock_t duration));

1828 /*
1829  * Fault-related functions
1830  */
1831 extern int ddi_check_acc_handle(ddi_acc_handle_t);
1832 extern int ddi_check_dma_handle(ddi_dma_handle_t);
1833 extern void ddi_dev_report_fault(dev_info_t *, ddi_fault_impact_t,
1834         ddi_fault_location_t, const char *);
1835 extern ddi_devstate_t ddi_get_devstate(dev_info_t *);

1837 /*
1838  * Miscellaneous redefines
1839  */
1840 #define uiophysio       physio

1842 /*
1843  * utilities - "reg" mapping and all common portable data access functions
1844  */

1846 /*
1847  * error code from ddi_regs_map_setup
1848  */

1850 #define DDI_REGS_ACC_CONFLICT   (-10)

1852 /*
1853  * Device address advance flags
1854  */

1856 #define  DDI_DEV_NO_AUTOINCR    0x0000
1857 #define  DDI_DEV_AUTOINCR       0x0001

1859 int
1860 ddi_regs_map_setup(dev_info_t *dip, uint_t rnumber, caddr_t *addrp,
```



```
1861         offset_t offset, offset_t len, ddi_device_acc_attr_t *accattrp,
1862         ddi_acc_handle_t *handle);

1864 void
1865 ddi_regs_map_free(ddi_acc_handle_t *handle);

1867 /*
1868  * these are the prototypes for the common portable data access functions
1869  */

1871 uint8_t
1872 ddi_get8(ddi_acc_handle_t handle, uint8_t *addr);

1874 uint16_t
1875 ddi_get16(ddi_acc_handle_t handle, uint16_t *addr);

1877 uint32_t
1878 ddi_get32(ddi_acc_handle_t handle, uint32_t *addr);

1880 uint64_t
1881 ddi_get64(ddi_acc_handle_t handle, uint64_t *addr);

1883 void
1884 ddi_rep_get8(ddi_acc_handle_t handle, uint8_t *host_addr, uint8_t *dev_addr,
1885         size_t repcount, uint_t flags);

1887 void
1888 ddi_rep_get16(ddi_acc_handle_t handle, uint16_t *host_addr, uint16_t *dev_addr,
1889         size_t repcount, uint_t flags);

1891 void
1892 ddi_rep_get32(ddi_acc_handle_t handle, uint32_t *host_addr, uint32_t *dev_addr,
1893         size_t repcount, uint_t flags);

1895 void
1896 ddi_rep_get64(ddi_acc_handle_t handle, uint64_t *host_addr, uint64_t *dev_addr,
1897         size_t repcount, uint_t flags);

1899 void
1900 ddi_put8(ddi_acc_handle_t handle, uint8_t *addr, uint8_t value);

1902 void
1903 ddi_put16(ddi_acc_handle_t handle, uint16_t *addr, uint16_t value);

1905 void
1906 ddi_put32(ddi_acc_handle_t handle, uint32_t *addr, uint32_t value);

1908 void
1909 ddi_put64(ddi_acc_handle_t handle, uint64_t *addr, uint64_t value);

1911 void
1912 ddi_rep_put8(ddi_acc_handle_t handle, uint8_t *host_addr, uint8_t *dev_addr,
1913         size_t repcount, uint_t flags);
1914 void
1915 ddi_rep_put16(ddi_acc_handle_t handle, uint16_t *host_addr, uint16_t *dev_addr,
1916         size_t repcount, uint_t flags);
1917 void
1918 ddi_rep_put32(ddi_acc_handle_t handle, uint32_t *host_addr, uint32_t *dev_addr,
1919         size_t repcount, uint_t flags);

1921 void
1922 ddi_rep_put64(ddi_acc_handle_t handle, uint64_t *host_addr, uint64_t *dev_addr,
1923         size_t repcount, uint_t flags);

1925 /*
1926  * these are special device handling functions
```

```
 */
int
ddi_device_zero(ddi_acc_handle_t handle, caddr_t dev_addr,
        size_t bytecount, ssize_t dev_advcnt, uint_t dev_datasz);

int
ddi_device_copy(
        ddi_acc_handle_t src_handle, caddr_t src_addr, ssize_t src_advcnt,
        ddi_acc_handle_t dest_handle, caddr_t dest_addr, ssize_t dest_advcnt,
        size_t bytecount, uint_t dev_datasz);

/*
 * these are software byte swapping functions
 */
uint16_t
ddi_swap16(uint16_t value);

uint32_t
ddi_swap32(uint32_t value);

uint64_t
ddi_swap64(uint64_t value);

/*
 * these are the prototypes for PCI local bus functions
 */
/*
 * PCI power management capabilities reporting in addition to those
 * provided by the PCI Power Management Specification.
 */
#define PCI_PM_IDLESPEED        0x1             /* clock for idle dev - cap  */
#define PCI_PM_IDLESPEED_ANY    (void *)-1      /* any clock for idle dev */
#define PCI_PM_IDLESPEED_NONE   (void *)-2      /* regular clock for idle dev */

int
pci_config_setup(dev_info_t *dip, ddi_acc_handle_t *handle);

void
pci_config_teardown(ddi_acc_handle_t *handle);

uint8_t
pci_config_get8(ddi_acc_handle_t handle, off_t offset);

uint16_t
pci_config_get16(ddi_acc_handle_t handle, off_t offset);

uint32_t
pci_config_get32(ddi_acc_handle_t handle, off_t offset);

uint64_t
pci_config_get64(ddi_acc_handle_t handle, off_t offset);

void
pci_config_put8(ddi_acc_handle_t handle, off_t offset, uint8_t value);

void
pci_config_put16(ddi_acc_handle_t handle, off_t offset, uint16_t value);

void
pci_config_put32(ddi_acc_handle_t handle, off_t offset, uint32_t value);

void
pci_config_put64(ddi_acc_handle_t handle, off_t offset, uint64_t value);

int
pci_report_pmcap(dev_info_t *dip, int cap, void *arg);
```



```
int
pci_restore_config_regs(dev_info_t *dip);

int
pci_save_config_regs(dev_info_t *dip);

void
pci_ereport_setup(dev_info_t *dip);

void
pci_ereport_teardown(dev_info_t *dip);

void
pci_ereport_post(dev_info_t *dip, ddi_fm_error_t *derr, uint16_t *status);

#if defined(__i386) || defined(__amd64)
int
pci_peekpoke_check(dev_info_t *, dev_info_t *, ddi_ctl_enum_t, void *, void *,
        int (*handler)(dev_info_t *, dev_info_t *, ddi_ctl_enum_t, void *,
        void *), kmutex_t *, kmutex_t *,
        void (*scan)(dev_info_t *, ddi_fm_error_t *));
#endif

void
pci_target_enqueue(uint64_t, char *, char *, uint64_t);

void
pci_targetq_init(void);

int
pci_post_suspend(dev_info_t *dip);

int
pci_pre_resume(dev_info_t *dip);

/*
 * the prototype for the C Language Type Model inquiry.
 */
model_t ddi_mmap_get_model(void);
model_t ddi_model_convert_from(model_t);

/*
 * these are the prototypes for device id functions.
 */
int
ddi_devid_valid(ddi_devid_t devid);

int
ddi_devid_register(dev_info_t *dip, ddi_devid_t devid);

void
ddi_devid_unregister(dev_info_t *dip);

int
ddi_devid_init(dev_info_t *dip, ushort_t devid_type, ushort_t nbytes,
    void *id, ddi_devid_t *ret_devid);

int
ddi_devid_get(dev_info_t *dip, ddi_devid_t *ret_devid);

size_t
ddi_devid_sizeof(ddi_devid_t devid);

void
ddi_devid_free(ddi_devid_t devid);
```

```
2060 int
2061 ddi_devid_compare(ddi_devid_t id1, ddi_devid_t id2);

2063 int
2064 ddi_devid_scsi_encode(int version, char *driver_name,
2065     uchar_t *inq, size_t inq_len, uchar_t *inq80, size_t inq80_len,
2066     uchar_t *inq83, size_t inq83_len, ddi_devid_t *ret_devid);

2068 int
2069 ddi_devid_smp_encode(int version, char *driver_name,
2070     char *wwnstr, uchar_t *srmir_buf, size_t srmir_len,
2071     ddi_devid_t *ret_devid);

2073 char
2074 *ddi_devid_to_guid(ddi_devid_t devid);

2076 void
2077 ddi_devid_free_guid(char *guid);

2079 int
2080 ddi_lyr_get_devid(dev_t dev, ddi_devid_t *ret_devid);

2082 int
2083 ddi_lyr_get_minor_name(dev_t dev, int spec_type, char **minor_name);

2085 int
2086 ddi_lyr_devid_to_devlist(ddi_devid_t devid, char *minor_name, int *retndevs,
2087     dev_t **retdevs);

2089 void
2090 ddi_lyr_free_devlist(dev_t *devlist, int ndevs);

2092 char *
2093 ddi_devid_str_encode(ddi_devid_t devid, char *minor_name);

2095 int
2096 ddi_devid_str_decode(char *devidstr, ddi_devid_t *devidp, char **minor_namep);

2098 void
2099 ddi_devid_str_free(char *devidstr);

2101 int
2102 ddi_devid_str_compare(char *id1_str, char *id2_str);

2104 /*
2105  * Event to post to when a devinfo node is removed.
2106  */
2107 #define DDI_DEVI_REMOVE_EVENT           "DDI:DEVI_REMOVE"
2108 #define DDI_DEVI_INSERT_EVENT           "DDI:DEVI_INSERT"
2109 #define DDI_DEVI_BUS_RESET_EVENT        "DDI:DEVI_BUS_RESET"
2110 #define DDI_DEVI_DEVICE_RESET_EVENT     "DDI:DEVI_DEVICE_RESET"

2112 /*
2113  * Invoke bus nexus driver's implementation of the
2114  * (*bus_remove_eventcall)() interface to remove a registered
2115  * callback handler for "event".
2116  */
2117 int
2118 ddi_remove_event_handler(ddi_callback_id_t id);

2120 /*
2121  * Invoke bus nexus driver's implementation of the
2122  * (*bus_add_eventcall)() interface to register a callback handler
2123  * for "event".
2124  */
```



```
2125 int
2126 ddi_add_event_handler(dev_info_t *dip, ddi_eventcookie_t event,
2127         void (*handler)(dev_info_t *, ddi_eventcookie_t, void *, void *),
2128         void *arg, ddi_callback_id_t *id);

2130 /*
2131  * Return a handle for event "name" by calling up the device tree
2132  * hierarchy via  (*bus_get_eventcookie)() interface until claimed
2133  * by a bus nexus or top of dev_info tree is reached.
2134  */
2135 int
2136 ddi_get_eventcookie(dev_info_t *dip, char *name,
2137         ddi_eventcookie_t *event_cookiep);

2139 /*
2140  * log a system event
2141  */
2142 int
2143 ddi_log_sysevent(dev_info_t *dip, char *vendor, char *class_name,
2144         char *subclass_name, nvlist_t *attr_list, sysevent_id_t *eidp,
2145         int sleep_flag);

2147 /*
2148  * ddi_log_sysevent() vendors
2149  */
2150 #define DDI_VENDOR_SUNW         "SUNW"

2152 /*
2153  * Opaque task queue handle.
2154  */
2155 typedef struct ddi_taskq ddi_taskq_t;

2157 /*
2158  * Use default system priority.
2159  */
2160 #define TASKQ_DEFAULTPRI -1

2162 /*
2163  * Create a task queue
2164  */
2165 ddi_taskq_t *ddi_taskq_create(dev_info_t *dip, const char *name,
2166         int nthreads, pri_t pri, uint_t cflags);

2168 /*
2169  * destroy a task queue
2170  */
2171 void ddi_taskq_destroy(ddi_taskq_t *tq);

2173 /*
2174  * Dispatch a task to a task queue
2175  */
2176 int ddi_taskq_dispatch(ddi_taskq_t *tq, void (* func)(void *),
2177         void *arg, uint_t dflags);

2179 /*
2180  * Wait for all previously scheduled tasks to complete.
2181  */
2182 void ddi_taskq_wait(ddi_taskq_t *tq);

2184 /*
2185  * Suspend all task execution.
2186  */
2187 void ddi_taskq_suspend(ddi_taskq_t *tq);

2189 /*
2190  * Resume task execution.
```

```
 */
void ddi_taskq_resume(ddi_taskq_t *tq);

/*
 * Is task queue suspended?
 */
boolean_t ddi_taskq_suspended(ddi_taskq_t *tq);

/*
 * Parse an interface name of the form <alphanumeric>##<numeric> where
 * <numeric> is maximal.
 */
int ddi_parse(const char *, char *, uint_t *);

/*
 * DDI interrupt priority level
 */
#define DDI_IPL_0       (0)     /* kernel context */
#define DDI_IPL_1       (1)     /* interrupt priority level 1 */
#define DDI_IPL_2       (2)     /* interrupt priority level 2 */
#define DDI_IPL_3       (3)     /* interrupt priority level 3 */
#define DDI_IPL_4       (4)     /* interrupt priority level 4 */
#define DDI_IPL_5       (5)     /* interrupt priority level 5 */
#define DDI_IPL_6       (6)     /* interrupt priority level 6 */
#define DDI_IPL_7       (7)     /* interrupt priority level 7 */
#define DDI_IPL_8       (8)     /* interrupt priority level 8 */
#define DDI_IPL_9       (9)     /* interrupt priority level 9 */
#define DDI_IPL_10      (10)    /* interrupt priority level 10 */

/*
 * DDI periodic timeout interface
 */
ddi_periodic_t ddi_periodic_add(void (*)(void *), void *, hrtime_t, int);
void ddi_periodic_delete(ddi_periodic_t);

/*
 * Default quiesce(9E) implementation for drivers that don't need to do
 * anything.
 */
int ddi_quiesce_not_needed(dev_info_t *);

/*
 * Default quiesce(9E) initialization function for drivers that should
 * implement quiesce but haven't yet.
 */
int ddi_quiesce_not_supported(dev_info_t *);

/*
 * DDI generic callback interface
 */

typedef struct __ddi_cb **ddi_cb_handle_t;

int     ddi_cb_register(dev_info_t *dip, ddi_cb_flags_t flags,
            ddi_cb_func_t cbfunc, void *arg1, void *arg2,
            ddi_cb_handle_t *ret_hdlp);
int     ddi_cb_unregister(ddi_cb_handle_t hdl);

/* Notify DDI of memory added */
void ddi_mem_update(uint64_t addr, uint64_t size);

/* Path alias interfaces */
typedef struct plat_alias {
        char *pali_current;
        uint64_t pali_naliases;
        char **pali_aliases;
```



```
} plat_alias_t;
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   44929 Wed Feb 26 14:23:25 2014
new/usr/src/uts/intel/io/dktp/dcdev/dadk.c
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
**********************************************************
______unchanged_portion_omitted_


1029 tgdk_iob_handle
1030 dadk_iob_alloc(opaque_t objp, daddr_t blkno, ssize_t xfer, int kmsflg)
1031 {
1032         struct dadk *dadkp = (struct dadk *)objp;
1033         struct buf *bp;
1034         struct tgdk_iob *iobp;
1035         size_t rlen;

1037         iobp = kmem_zalloc(sizeof (*iobp), kmsflg);
1038         if (iobp == NULL)
1039                 return (NULL);
1040         if ((bp = getrbuf(kmsflg)) == NULL) {
1041                 kmem_free(iobp, sizeof (*iobp));
1042                 return (NULL);
1043         }

1045         iobp->b_psec  = LBLK2SEC(blkno, dadkp->dad_blkshf);
1046         iobp->b_pbyteoff = (blkno & ((1<<dadkp->dad_blkshf) - 1)) << SCTRSHFT;
1047         iobp->b_pbytecnt = ((iobp->b_pbyteoff + xfer + dadkp->DAD_SECSIZ - 1)
1048             >> dadkp->dad_secshf) << dadkp->dad_secshf;

1050         bp->b_un.b_addr = 0;
1051         /*
1052          * use i_ddi_mem_alloc() for now until we have an interface to allocate
1053          * memory for DMA which doesn't require a DMA handle.
1053          * memory for DMA which doesn't require a DMA handle. ddi_iopb_alloc()
1054          * is obsolete and we want more flexibility in controlling the DMA
1055          * address constraints..
1054          */
1055         if (i_ddi_mem_alloc((dadkp->dad_sd)->sd_dev, &dadk_alloc_attr,
1056             (size_t)iobp->b_pbytecnt, ((kmsflg == KM_SLEEP) ? 1 : 0), 0, NULL,
1057             &bp->b_un.b_addr, &rlen, NULL) != DDI_SUCCESS) {
1058                 freerbuf(bp);
1059                 kmem_free(iobp, sizeof (*iobp));
1060                 return (NULL);
1061         }
1062         iobp->b_flag |= IOB_BPALLOC | IOB_BPBUFALLOC;
1063         iobp->b_bp = bp;
1064         iobp->b_lblk = blkno;
1065         iobp->b_xfer = xfer;
1066         iobp->b_lblk = blkno;
1067         iobp->b_xfer = xfer;
1068         return (iobp);
1069 }
______unchanged_portion_omitted_
```