
27182 Sat May 24 17:48:25 2014

new/usr/src/man/man9f/Intro.9f

4888 Undocument dma_req(9s)

4884 EOF scsi_hba_attach

4886 EOF ddi_dmae_getlim

4887 EOF ddi_iomin

4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)

4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free

```

1  \" te
2  \". Copyright 2014 Garrett D'Amore <garrett@damore.org>
3  \". 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  \". TH INTRO 9F "Feb 06, 2012"
8  .SH NAME
9  Intro, intro \- introduction to DDI/DKI functions
10 .SH DESCRIPTION
11 .sp
12 .LP
13 Section 9F describes the kernel functions available for use by device drivers.
14 See \fBIntro\fR(9E) for an overview of device driver interfaces.
15 .sp
16 .LP
17 In this section, the information for each driver function is organized under
18 the following headings:
19 .RS +4
20 .TP
21 .ie t \(\bu
22 .el o
23 \fBNAME\fR summarizes the function's purpose.
24 .RE
25 .RS +4
26 .TP
27 .ie t \(\bu
28 .el o
29 \fBSYNOPSIS\fR shows the syntax of the function's entry point in the source
30 code. \fB#include\fR directives are shown for required headers.
31 .RE
32 .RS +4
33 .TP
34 .ie t \(\bu
35 .el o
36 \fBINTERFACE\fR \fBLEVEL\fR describes any architecture dependencies.
37 .RE
38 .RS +4
39 .TP
40 .ie t \(\bu
41 .el o
42 \fBARGUMENTS\fR describes any arguments required to invoke the function.
43 .RE
44 .RS +4
45 .TP
46 .ie t \(\bu
47 .el o
48 \fBDESCRIPTION\fR describes general information about the function.
49 .RE
50 .RS +4
51 .TP
52 .ie t \(\bu
53 .el o
54 \fBRETURN\fR \fBVALUES\fR describes the return values and messages that can
55 result from invoking the function.

```

```

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

```

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

132 .sp
 133 .TS
 134 c c
 135 l l .
 136 Routine Type
 137 -
 138 \fBbadjmsg\fR DDI/DKI
 139 \fBballocb\fR DDI/DKI
 140 \fBballocb_tmpl\fR Solaris DDI
 141 \fBbackq\fR DDI/DKI
 142 \fBbcinput\fR DDI/DKI
 143 \fBbcinputnext\fR DDI/DKI
 144 \fBbufcall\fR DDI/DKI
 145 \fBcanput\fR DDI/DKI
 146 \fBcanputnext\fR DDI/DKI
 147 \fBclrbuf\fR DDI/DKI
 148 \fBcopyb\fR DDI/DKI
 149 \fBcopymsg\fR DDI/DKI
 150 \fBBDB_BASE\fR Solaris DDI
 151 \fBBDB_LIM\fR Solaris DDI
 152 \fBBDB_REF\fR Solaris DDI
 153 \fBBDB_TYPE\fR Solaris DDI
 154 \fBdatamsg\fR DDI/DKI
 155 \fBdupb\fR DDI/DKI
 156 \fBdupmsg\fR DDI/DKI
 157 \fBenableok\fR DDI/DKI
 158 \fBesballocc\fR DDI/DKI
 159 \fBesbbcall\fR DDI/DKI
 160 \fBflushband\fR DDI/DKI
 161 \fBflushq\fR DDI/DKI
 162 \fBfreeb\fR DDI/DKI
 163 \fBfreemsg\fR DDI/DKI
 164 \fBfreezestr\fR DDI/DKI
 165 \fBgetq\fR DDI/DKI
 166 \fBIOC_CONVER_FROM\fR Solaris DDI
 167 \fBbinsq\fR DDI/DKI
 168 \fBlinkb\fR DDI/DKI
 169 \fBMBLKHEAD\fR Solaris DDI
 170 \fBMBLKIN\fR Solaris DDI
 171 \fBMBLKL\fR Solaris DDI
 172 \fBMBLKSIZEL\fR Solaris DDI
 173 \fBMBLKTAIL\fR Solaris DDI
 174 \fBmcopyin\fR Solaris DDI
 175 \fBmcopymsg\fR Solaris DDI
 176 \fBmcopyout\fR Solaris DDI
 177 \fBmerror\fR Solaris DDI
 178 \fBmexchange\fR Solaris DDI
 179 \fBmioc2ack\fR Solaris DDI
 180 \fBmiocack\fR Solaris DDI
 181 \fBmexchange\fR Solaris DDI
 182 \fBmiocpullup\fR Solaris DDI
 183 \fBmkiocb\fR Solaris DDI
 184 \fBmsgsize\fR DDI/DKI
 185 \fBmsgpullup\fR DDI/DKI
 186 \fBmsgsize\fR Solaris DDI
 187 \fBmt-streams\fR Solaris DDI

188 \fBnoenable\fR DDI/DKI
 189 \fBOTHERQ\fR DDI/DKI
 190 \fBpullupmsg\fR DDI/DKI
 191 \fBput\fR DDI/DKI
 192 \fBputbq\fR DDI/DKI
 193 \fBputctl\fR DDI/DKI
 194 \fBputctl1\fR DDI/DKI
 195 \fBputnext\fR DDI/DKI
 196 \fBputnextctl\fR DDI/DKI
 197 \fBputq\fR DDI/DKI
 198 \fBqassociate\fR Solaris DDI
 199 \fBqbufcall\fR Solaris DDI
 200 \fBqenable\fR DDI/DKI
 201 \fBqprocson\fR DDI/DKI
 202 \fBqprocsoff\fR DDI/DKI
 203 \fBqreply\fR DDI/DKI
 204 \fBqsize\fR DDI/DKI
 205 \fBqtimeout\fR Solaris DDI
 206 \fBqunbufcall\fR Solaris DDI
 207 \fBquntimeout\fR Solaris DDI
 208 \fBqwait\fR Solaris DDI
 209 \fBqwait_sig\fR Solaris DDI
 210 \fBqwriter\fR Solaris DDI
 211 \fBRD\fR DDI/DKI
 212 \fBrmvb\fR DDI/DKI
 213 \fBrmvq\fR DDI/DKI
 214 \fBSAMESTR\fR DDI/DKI
 215 \fBstrlog\fR DDI/DKI
 216 \fBstrqget\fR DDI/DKI
 217 \fBstrqset\fR DDI/DKI
 218 \fBtestb\fR DDI/DKI
 219 \fBunbufcall\fR DDI/DKI
 220 \fBunfreezestr\fR DDI/DKI
 221 \fBunlinkb\fR DDI/DKI
 222 \fBWR\fR DDI/DKI
 223 .TE
 225 .sp
 226 .LP
 227 The following table summarizes the functions not specific to STREAMS.
 228 .sp
 230 .sp
 231 .TS
 232 c c
 233 l l .
 234 Routine Type
 235 -
 236 \fBASSERT\fR DDI/DKI
 237 \fBAnocancel\fR Solaris DDI
 238 \fBbaphysio\fR Solaris DDI
 239 \fBatomic_add\fR DDI/DKI
 240 \fBatomic_and\fR DDI/DKI
 241 \fBatomic_bits\fR DDI/DKI
 242 \fBatomic_cas\fR DDI/DKI
 243 \fBatomic_dec\fR DDI/DKI
 244 \fBatomic_inc\fR DDI/DKI
 245 \fBatomic_ops\fR DDI/DKI
 246 \fBatomic_or\fR DDI/DKI
 247 \fBatomic_swap\fR DDI/DKI
 248 \fBbcmp\fR DDI/DKI
 249 \fBbcopy\fR DDI/DKI
 250 \fBbioclone\fR Solaris DDI
 251 \fBbiodone\fR DDI/DKI
 252 \fBbiofini\fR Solaris DDI
 253 \fBbioinit\fR Solaris DDI

```

254 \fBbiomodified\fR Solaris DDI
255 \fBbiosize\fR Solaris DDI
256 \fBbioerror\fR Solaris DDI
257 \fBbioreset\fR Solaris DDI
258 \fBbiowait\fR DDI/DKI
259 \fBbbp_copyin\fR DDI/DKI
260 \fBbbp_copyout\fR DDI/DKI
261 \fBbbp_mapin\fR DDI/DKI
262 \fBbbp_mapout\fR DDI/DKI
263 \fBbktop\fR DDI/DKI
264 \fBbktopr\fR DDI/DKI
265 \fBbzero\fR DDI/DKI
266 \fBbmm_err\fR DDI/DKI
267 \fBbcondvar\fR Solaris DDI
268 \fBbcopyin\fR DDI/DKI
269 \fBbcopyout\fR DDI/DKI
270 \fBbcxs_AccessConfigurationRegister\fR Solaris DDI
271 \fBbcxs_ConvertSize\fR Solaris DDI
272 \fBbcxs_ConvertSpeed\fR Solaris DDI
273 \fBbcxs_CS_DDI_Info\fR Solaris DDI
274 \fBbcxs_DeregisterClient\fR Solaris DDI
275 \fBbcxs_DupHandle\fR Solaris DDI
276 \fBbcxs_Error2Text\fR Solaris DDI
277 \fBbcxs_Event2Text\fR Solaris DDI
278 \fBbcxs_FreeHandle\fR Solaris DDI
279 \fBbcxs_Get8\fR Solaris DDI
280 \fBbcxs_GetFirstClient\fR Solaris DDI
281 \fBbcxs_GetFirstTuple\fR Solaris DDI
282 \fBbcxs_GetHandleOffset\fR Solaris DDI
283 \fBbcxs_GetMappedAddr\fR Solaris DDI
284 \fBbcxs_GetStatus\fR Solaris DDI
285 \fBbcxs_GetTupleData\fR Solaris DDI
286 \fBbcxs_MakeDeviceNode\fR Solaris DDI
287 \fBbcxs_MapLogSocket\fR Solaris DDI
288 \fBbcxs_MapMemPage\fR Solaris DDI
289 \fBbcxs_ModifyConfiguration\fR Solaris DDI
290 \fBbcxs_ModifyWindow\fR Solaris DDI
291 \fBbcxs_Parse_CISTPL_BATTERY\fR Solaris DDI
292 \fBbcxs_Parse_CISTPL_BYTEORDER\fR Solaris DDI
293 \fBbcxs_Parse_CISTPL_CFTABLE_ENTRY\fR Solaris DDI
294 \fBbcxs_Parse_CISTPL_CONFIG\fR Solaris DDI
295 \fBbcxs_Parse_CISTPL_DATE\fR Solaris DDI
296 \fBbcxs_Parse_CISTPL_DEVICE\fR Solaris DDI
297 \fBbcxs_Parse_CISTPL_DEVICEGEO\fR Solaris DDI
298 \fBbcxs_Parse_CISTPL_DEVICEGEO_A\fR Solaris DDI
299 \fBbcxs_Parse_CISTPL_FORMAT\fR Solaris DDI
300 \fBbcxs_Parse_CISTPL_FUNCCE\fR Solaris DDI
301 \fBbcxs_Parse_CISTPL_FUNCID\fR Solaris DDI
302 \fBbcxs_Parse_CISTPL_GEOMETRY\fR Solaris DDI
303 \fBbcxs_Parse_CISTPL_JEDEC_C\fR Solaris DDI
304 \fBbcxs_Parse_CISTPL_LINKTARGET\fR Solaris DDI
305 \fBbcxs_Parse_CISTPL_LONGLINK_A\fR Solaris DDI
306 \fBbcxs_Parse_CISTPL_LONGLINK_MFC\fR Solaris DDI
307 \fBbcxs_Parse_CISTPL_MANFID\fR Solaris DDI
308 \fBbcxs_Parse_CISTPL_ORG\fR Solaris DDI
309 \fBbcxs_Parse_CISTPL_SPCL\fR Solaris DDI
310 \fBbcxs_Parse_CISTPL_SWIL\fR Solaris DDI
311 \fBbcxs_Parse_CISTPL_VERS_1\fR Solaris DDI
312 \fBbcxs_Parse_CISTPL_VERS_2\fR Solaris DDI
313 \fBbcxs_ParseTuple\fR Solaris DDI
314 \fBbcxs_Put8\fR Solaris DDI
315 \fBbcxs_RegisterClient\fR Solaris DDI
316 \fBbcxs_ReleaseConfiguration\fR Solaris DDI
317 \fBbcxs_RepGet8\fR Solaris DDI
318 \fBbcxs_RepPut8\fR Solaris DDI
319 \fBbcxs_RequestConfiguration\fR Solaris DDI

```

```

320 \fBbcxs_RequestIO\fR Solaris DDI
321 \fBbcxs_RequestIRQ\fR Solaris DDI
322 \fBbcxs_RequestSocketMask\fR Solaris DDI
323 \fBbcxs_RequestWindow\fR Solaris DDI
324 \fBbcxs_ResetFunction\fR Solaris DDI
325 \fBbcxs_SetEventMask\fR Solaris DDI
326 \fBbcxs_SetHandleOffset\fR Solaris DDI
327 \fBbcxs_ValidateCIS\fR Solaris DDI
328 \fBbcv_broadcast\fR Solaris DDI
329 \fBbcv_destroy\fR Solaris DDI
330 \fBbcv_init\fR Solaris DDI
331 \fBbcv_signal\fR Solaris DDI
332 \fBbcv_timedwait\fR Solaris DDI
333 \fBbcv_wait\fR Solaris DDI
334 \fBbcv_wait_sig\fR Solaris DDI
335 \fBbddi_add_event_handler\fR Solaris DDI
336 \fBbddi_add_intr\fR Solaris DDI
337 \fBbddi_add_softintr\fR Solaris DDI
338 \fBbddi_binding_name\fR Solaris DDI
339 \fBbddi_btop\fR Solaris DDI
340 \fBbddi_btopr\fR Solaris DDI
341 \fBbddi_can_receive_sig\fR Solaris DDI
342 \fBbddi_check_acc_handle\fR Solaris DDI
343 \fBbddi_copyin\fR Solaris DDI
344 \fBbddi_copyout\fR Solaris DDI
345 \fBbddi_create_minor_node\fR Solaris DDI
346 \fBbddi_cred\fR Solaris DDI
347 \fBbddi_dev_is_sid\fR Solaris DDI
348 \fBbddi_dev_nintrs\fR Solaris DDI
349 \fBbddi_dev_nregs\fR Solaris DDI
350 \fBbddi_dev_regsizes\fR Solaris DDI
351 \fBbddi_device_copy\fR Solaris DDI
352 \fBbddi_device_zero\fR Solaris DDI
353 \fBbddi_devmap_segmap\fR Solaris DDI
354 \fBbddi_dma_addr_bind_handle\fR Solaris DDI
355 \fBbddi_dma_alloc_handle\fR Solaris DDI
356 \fBbddi_dma_buf_bind_handle\fR Solaris DDI
357 \fBbddi_dma_burstsizes\fR Solaris DDI
358 \fBbddi_dma_free_handle\fR Solaris DDI
359 \fBbddi_dma_getwin\fR Solaris DDI
360 \fBbddi_dma_mem_alloc\fR Solaris DDI
361 \fBbddi_dma_mem_free\fR Solaris DDI
362 \fBbddi_dma_nextcookie\fR Solaris DDI
363 \fBbddi_dma_numwin\fR Solaris DDI
364 \fBbddi_dma_set_sbus64\fR Solaris DDI
365 \fBbddi_dma_sync\fR Solaris DDI
366 \fBbddi_dma_unbind_handle\fR Solaris DDI
367 \fBbddi_dmae\fR Solaris x86 DDI
368 \fBbddi_dmae_1stparty\fR Solaris x86 DDI
369 \fBbddi_dmae_alloc\fR Solaris x86 DDI
370 \fBbddi_dmae_disable\fR Solaris x86 DDI
371 \fBbddi_dmae_enable\fR Solaris x86 DDI
372 \fBbddi_dmae_getattr\fR Solaris x86 DDI
373 \fBbddi_dmae_getcnt\fR Solaris x86 DDI
374 \fBbddi_dmae_getlim\fR Solaris x86 DDI
375 \fBbddi_dmae_prog\fR Solaris x86 DDI
376 \fBbddi_dmae_release\fR Solaris x86 DDI
377 \fBbddi_dmae_stop\fR Solaris x86 DDI
378 \fBbddi_driver_major\fR Solaris DDI
379 \fBbddi_driver_name\fR Solaris DDI
380 \fBbddi_enter_critical\fR Solaris DDI
381 \fBbddi_exit_critical\fR Solaris DDI
382 \fBbddi_ffs\fR Solaris DDI
383 \fBbddi_fls\fR Solaris DDI
384 \fBbddi_fm_acc_err_clear\fR Solaris DDI
385 \fBbddi_fm_acc_err_get\fR Solaris DDI

```

```

386 \fBddi_fm_ereport_post\fR      Solaris DDI
387 \fBddi_fm_handler_register\fR   Solaris DDI
388 \fBddi_fm_init\fR                Solaris DDI
389 \fBddi_fm_service_impact\fR     Solaris DDI
390 \fBddi_get16\fR                   Solaris DDI
391 \fBddi_get32\fR                   Solaris DDI
392 \fBddi_get64\fR                   Solaris DDI
393 \fBddi_get8\fR                    Solaris DDI
394 \fBddi_get_cred\fR               Solaris DDI
395 \fBddi_get_devstate\fR           Solaris DDI
396 \fBddi_get_driver_private\fR     Solaris DDI
397 \fBddi_get_eventcookie\fR        Solaris DDI
398 \fBddi_get_iblock_cookie\fR      Solaris DDI
399 \fBddi_get_iminor\fR             Solaris DDI
400 \fBddi_get_instance\fR           Solaris DDI
401 \fBddi_get_kt_did\fR            Solaris DDI
402 \fBddi_get_lbolt\fR             Solaris DDI
403 \fBddi_get_name\fR              Solaris DDI
404 \fBddi_get_parent\fR            Solaris DDI
405 \fBddi_get_pid\fR               Solaris DDI
406 \fBddi_get_soft_iblock_cookie\fR Solaris DDI
407 \fBddi_get_soft_state\fR        Solaris DDI
408 \fBddi_getb\fR                   Solaris DDI
409 \fBddi_getl\fR                   Solaris DDI
410 \fBddi_getll\fR                  Solaris DDI
411 \fBddi_getlongprop\fR           Solaris DDI
412 \fBddi_getlongprop_buf\fR       Solaris DDI
413 \fBddi_getprop\fR               Solaris DDI
414 \fBddi_getproplen\fR            Solaris DDI
415 \fBddi_getw\fR                   Solaris DDI
416 \fBddi_intr_add_handler\fR       Solaris DDI
417 \fBddi_intr_add_softint\fR       Solaris DDI
418 \fBddi_intr_alloc\fR            Solaris DDI
419 \fBddi_intr_block_disable\fR     Solaris DDI
420 \fBddi_intr_block_enable\fR     Solaris DDI
421 \fBddi_intr_clr_mask\fR          Solaris DDI
422 \fBddi_intr_dup_handler\fR       Solaris DDI
423 \fBddi_intr_disable\fR           Solaris DDI
424 \fBddi_intr_enable\fR           Solaris DDI
425 \fBddi_intr_free\fR             Solaris DDI
426 \fBddi_intr_get_cap\fR           Solaris DDI
427 \fBddi_intr_get_hilevel_pri\fR   Solaris DDI
428 \fBddi_intr_get_navail\fR        Solaris DDI
429 \fBddi_intr_get_nintrs\fR        Solaris DDI
430 \fBddi_intr_get_pending\fR       Solaris DDI
431 \fBddi_intr_get_pri\fR           Solaris DDI
432 \fBddi_intr_get_softint_pri\fR   Solaris DDI
433 \fBddi_intr_get_supported_types\fR Solaris DDI
434 \fBddi_intr_remove_handler\fR     Solaris DDI
435 \fBddi_intr_remove_softint\fR     Solaris DDI
436 \fBddi_intr_set_cap\fR           Solaris DDI
437 \fBddi_intr_set_mask\fR          Solaris DDI
438 \fBddi_intr_set_pri\fR           Solaris DDI
439 \fBddi_intr_set_softint_pri\fR   Solaris DDI
440 \fBddi_intr_trigger_softint\fR   Solaris DDI
441 \fBddi_io_get16\fR               Solaris DDI
442 \fBddi_io_get32\fR              Solaris DDI
443 \fBddi_io_get8\fR                Solaris DDI
444 \fBddi_io_getb\fR                Solaris DDI
445 \fBddi_io_getl\fR                Solaris DDI
446 \fBddi_io_getw\fR                Solaris DDI
447 \fBddi_io_put16\fR               Solaris DDI
448 \fBddi_io_put32\fR               Solaris DDI
449 \fBddi_io_put8\fR                Solaris DDI
450 \fBddi_io_putb\fR                Solaris DDI
451 \fBddi_io_putl\fR                Solaris DDI

```

```

452 \fBddi_io_putw\fR               Solaris DDI
453 \fBddi_io_rep_get16\fR           Solaris DDI
454 \fBddi_io_rep_get32\fR          Solaris DDI
455 \fBddi_io_rep_get8\fR           Solaris DDI
456 \fBddi_io_rep_getb\fR           Solaris DDI
457 \fBddi_io_rep_getl\fR           Solaris DDI
458 \fBddi_io_rep_getw\fR           Solaris DDI
459 \fBddi_io_rep_put16\fR          Solaris DDI
460 \fBddi_io_rep_put32\fR          Solaris DDI
461 \fBddi_io_rep_put8\fR           Solaris DDI
462 \fBddi_io_rep_putb\fR           Solaris DDI
463 \fBddi_io_rep_putl\fR           Solaris DDI
464 \fBddi_io_rep_putw\fR           Solaris DDI
465 \fBddi_iomin\fR                 Solaris DDI
466 \fBddi_log_sysevent\fR          Solaris DDI
467 \fBddi_map_regs\fR              Solaris DDI
468 \fBddi_mapdev\fR                Solaris DDI
469 \fBddi_mapdev_intercept\fR       Solaris DDI
470 \fBddi_mapdev_nointercept\fR    Solaris DDI
471 \fBddi_mapdev_set_device_acc_attr\fR Solaris DDI
472 \fBddi_mem_get16\fR             Solaris DDI
473 \fBddi_mem_get32\fR             Solaris DDI
474 \fBddi_mem_get64\fR             Solaris DDI
475 \fBddi_mem_get8\fR              Solaris DDI
476 \fBddi_mem_getb\fR              Solaris DDI
477 \fBddi_mem_getl\fR              Solaris DDI
478 \fBddi_mem_getll\fR             Solaris DDI
479 \fBddi_mem_getw\fR              Solaris DDI
480 \fBddi_mem_put16\fR             Solaris DDI
481 \fBddi_mem_put32\fR             Solaris DDI
482 \fBddi_mem_put64\fR             Solaris DDI
483 \fBddi_mem_put8\fR              Solaris DDI
484 \fBddi_mem_putb\fR              Solaris DDI
485 \fBddi_mem_putl\fR              Solaris DDI
486 \fBddi_mem_putll\fR            Solaris DDI
487 \fBddi_mem_putw\fR              Solaris DDI
488 \fBddi_mem_rep_get16\fR          Solaris DDI
489 \fBddi_mem_rep_get32\fR          Solaris DDI
490 \fBddi_mem_rep_get64\fR          Solaris DDI
491 \fBddi_mem_rep_get8\fR           Solaris DDI
492 \fBddi_mem_rep_getb\fR           Solaris DDI
493 \fBddi_mem_rep_getl\fR           Solaris DDI
494 \fBddi_mem_rep_getll\fR          Solaris DDI
495 \fBddi_mem_rep_getw\fR           Solaris DDI
496 \fBddi_mem_rep_put16\fR          Solaris DDI
497 \fBddi_mem_rep_put32\fR          Solaris DDI
498 \fBddi_mem_rep_put64\fR          Solaris DDI
499 \fBddi_mem_rep_put8\fR           Solaris DDI
500 \fBddi_mem_rep_putb\fR           Solaris DDI
501 \fBddi_mem_rep_putl\fR           Solaris DDI
502 \fBddi_mem_rep_putll\fR          Solaris DDI
503 \fBddi_mem_rep_putw\fR           Solaris DDI
504 \fBddi_mmap_get_model\fR         Solaris DDI
505 \fBddi_model_convert_from\fR     Solaris DDI
506 \fBddi_modopen\fR                Solaris DDI
507 \fBddi_no_info\fR                Solaris DDI
508 \fBddi_node_name\fR              Solaris DDI
509 \fBddi_peek16\fR                 Solaris DDI
510 \fBddi_peek32\fR                 Solaris DDI
511 \fBddi_peek64\fR                 Solaris DDI
512 \fBddi_peek8\fR                  Solaris DDI
513 \fBddi_peekc\fR                  Solaris DDI
514 \fBddi_peekd\fR                  Solaris DDI
515 \fBddi_peekl\fR                  Solaris DDI
516 \fBddi_peeks\fR                  Solaris DDI
517 \fBddi_periodic_add\fR           Solaris DDI

```

518 \fBddi_periodic_delete\fR Solaris DDI
 519 \fBddi_poke16\fR Solaris DDI
 520 \fBddi_poke32\fR Solaris DDI
 521 \fBddi_poke64\fR Solaris DDI
 522 \fBddi_poke8\fR Solaris DDI
 523 \fBddi_pokec\fR Solaris DDI
 524 \fBddi_poked\fR Solaris DDI
 525 \fBddi_pokel\fR Solaris DDI
 526 \fBddi_pokes\fR Solaris DDI
 527 \fBddi_prop_create\fR Solaris DDI
 528 \fBddi_prop_exists\fR Solaris DDI
 529 \fBddi_prop_free\fR Solaris DDI
 530 \fBddi_prop_get_int\fR Solaris DDI
 531 \fBddi_prop_lookup\fR Solaris DDI
 532 \fBddi_prop_lookup_byte_array\fR Solaris DDI
 533 \fBddi_prop_lookup_int_array\fR Solaris DDI
 534 \fBddi_prop_lookup_string\fR Solaris DDI
 535 \fBddi_prop_lookup_string_array\fR Solaris DDI
 536 \fBddi_prop_modify\fR Solaris DDI
 537 \fBddi_prop_op\fR Solaris DDI
 538 \fBddi_prop_remove\fR Solaris DDI
 539 \fBddi_prop_remove_all\fR Solaris DDI
 540 \fBddi_prop_undefine\fR Solaris DDI
 541 \fBddi_prop_update\fR Solaris DDI
 542 \fBddi_prop_update_byte_array\fR Solaris DDI
 543 \fBddi_prop_update_int\fR Solaris DDI
 544 \fBddi_prop_update_int_array\fR Solaris DDI
 545 \fBddi_prop_update_string\fR Solaris DDI
 546 \fBddi_prop_update_string_array\fR Solaris DDI
 547 \fBddi_ptob\fR Solaris DDI
 548 \fBddi_put16\fR Solaris DDI
 549 \fBddi_put32\fR Solaris DDI
 550 \fBddi_put64\fR Solaris DDI
 551 \fBddi_put8\fR Solaris DDI
 552 \fBddi_putb\fR Solaris DDI
 553 \fBddi_putl\fR Solaris DDI
 554 \fBddi_putll\fR Solaris DDI
 555 \fBddi_putw\fR Solaris DDI
 556 \fBddi_regs_map_free\fR Solaris DDI
 557 \fBddi_regs_map_setup\fR Solaris DDI
 558 \fBddi_remove_event_handler\fR Solaris DDI
 559 \fBddi_remove_intr\fR Solaris DDI
 560 \fBddi_remove_minor_node\fR Solaris DDI
 561 \fBddi_remove_softintr\fR Solaris DDI
 562 \fBddi_removing_power\fR Solaris DDI
 563 \fBddi_rep_get16\fR Solaris DDI
 564 \fBddi_rep_get32\fR Solaris DDI
 565 \fBddi_rep_get64\fR Solaris DDI
 566 \fBddi_rep_get8\fR Solaris DDI
 567 \fBddi_rep_getb\fR Solaris DDI
 568 \fBddi_rep_getl\fR Solaris DDI
 569 \fBddi_rep_getll\fR Solaris DDI
 570 \fBddi_rep_getw\fR Solaris DDI
 571 \fBddi_rep_put16\fR Solaris DDI
 572 \fBddi_rep_put32\fR Solaris DDI
 573 \fBddi_rep_put64\fR Solaris DDI
 574 \fBddi_rep_put8\fR Solaris DDI
 575 \fBddi_rep_putb\fR Solaris DDI
 576 \fBddi_rep_putl\fR Solaris DDI
 577 \fBddi_rep_putll\fR Solaris DDI
 578 \fBddi_rep_putw\fR Solaris DDI
 579 \fBddi_report_dev\fR Solaris DDI
 580 \fBddi_root_node\fR Solaris DDI
 581 \fBddi_segmap\fR Solaris DDI
 582 \fBddi_segmap_setup\fR Solaris DDI
 583 \fBddi_set_driver_private\fR Solaris DDI

584 \fBddi_slaveonly\fR Solaris DDI
 585 \fBddi_soft_state\fR Solaris DDI
 586 \fBddi_soft_state_fini\fR Solaris DDI
 587 \fBddi_soft_state_free\fR Solaris DDI
 588 \fBddi_soft_state_init\fR Solaris DDI
 589 \fBddi_soft_state_zalloc\fR Solaris DDI
 590 \fBddi_strlol\fR Solaris DDI
 591 \fBddi_strloul\fR Solaris DDI
 592 \fBddi_trigger_softintr\fR Solaris DDI
 593 \fBddi_umem_alloc\fR Solaris DDI
 594 \fBddi_umem_free\fR Solaris DDI
 595 \fBddi_umem_iosetup\fR Solaris DDI
 596 \fBddi_umem_lock\fR Solaris DDI
 597 \fBddi_unmap_regs\fR Solaris DDI
 598 \fBdelay\fR DDI/DKI
 599 \fBdevmap_default_access\fR Solaris DDI
 600 \fBdevmap_devmem_setup\fR Solaris DDI
 601 \fBdevmap_do_ctxmgt\fR Solaris DDI
 602 \fBdevmap_load\fR Solaris DDI
 603 \fBdevmap_set_ctx_timeout\fR Solaris DDI
 604 \fBdevmap_setup\fR Solaris DDI
 605 \fBdevmap_umem_setup\fR Solaris DDI
 606 \fBdevmap_unload\fR Solaris DDI
 607 \fBdisksort\fR Solaris DDI
 608 \fBdlbindack\fR Solaris DDI
 609 \fBdrv_getparm\fR DDI/DKI
 610 \fBdrv_hztimeout\fR DDI/DKI
 611 \fBdrv_priv\fR DDI/DKI
 612 \fBdrv_usectohz\fR DDI/DKI
 613 \fBdrv_usecwait\fR DDI/DKI
 614 \fBfree_pktiopb\fR Solaris DDI
 615 \fBfreerbuf\fR DDI/DKI
 616 \fBget_pktiopb\fR Solaris DDI
 617 \fBgeterror\fR DDI/DKI
 618 \fBgethrtime\fR DDI/DKI
 619 \fBgetmajor\fR DDI/DKI
 620 \fBgetminor\fR DDI/DKI
 621 \fBgetrbuf\fR DDI/DKI
 622 \fBgld\fR Solaris DDI
 623 \fBhat_getkpfnum\fR DK1 only
 624 \fBid32_alloc\fR Solaris DDI
 625 \fBbinb\fR Solaris x86 DDI
 626 \fBbinl\fR Solaris x86 DDI
 627 \fBbinw\fR Solaris x86 DDI
 628 \fBkiconv\fR Solaris DDI
 629 \fBkiconv_close\fR Solaris DDI
 630 \fBkiconv_open\fR Solaris DDI
 631 \fBkiconvstr\fR Solaris DDI
 632 \fBkmem_alloc\fR DDI/DKI
 633 \fBkmem_cache_create\fR Solaris DDI
 634 \fBkmem_free\fR DDI/DKI
 635 \fBkmem_zalloc\fR DDI/DKI
 636 \fBkstat_create\fR Solaris DDI
 637 \fBkstat_delete\fR Solaris DDI
 638 \fBkstat_install\fR Solaris DDI
 639 \fBkstat_named_init\fR Solaris DDI
 640 \fBkstat_queue\fR Solaris DDI
 641 \fBkstat_runq_back_to_waitq\fR Solaris DDI
 642 \fBkstat_runq_enter\fR Solaris DDI
 643 \fBkstat_runq_exit\fR Solaris DDI
 644 \fBkstat_waitq_enter\fR Solaris DDI
 645 \fBkstat_waitq_exit\fR Solaris DDI
 646 \fBkstat_waitq_to_runq\fR Solaris DDI
 647 \fBldi_add_event_handler\fR Solaris DDI
 648 \fBldi_aread\fR Solaris DDI
 649 \fBldi_devmap\fR Solaris DDI

```

650 \fBldi_dump\fR Solaris DDI
651 \fBldi_ev_finalize\fR Solaris DDI
652 \fBldi_ev_get_cookie\fR Solaris DDI
653 \fBldi_ev_get_type\fR Solaris DDI
654 \fBldi_ev_notify\fR Solaris DDI
655 \fBldi_ev_register_callbacks\fR Solaris DDI
656 \fBldi_ev_remove_callbacks\fR Solaris DDI
657 \fBldi_get_dev\fR Solaris DDI
658 \fBldi_get_eventcookie\fR Solaris DDI
659 \fBldi_get_size\fR Solaris DDI
660 \fBldi_ident_from_dev\fR Solaris DDI
661 \fBldi_ioctl\fR Solaris DDI
662 \fBldi_open_by_dev\fR Solaris DDI
663 \fBldi_poll\fR Solaris DDI
664 \fBldi_prop_exists\fR Solaris DDI
665 \fBldi_prop_get_int\fR Solaris DDI
666 \fBldi_prop_get_lookup_int_array\fR Solaris DDI
667 \fBldi_putmsg\fR Solaris DDI
668 \fBldi_read\fR Solaris DDI
669 \fBldi_remove_event_handler\fR Solaris DDI
670 \fBldi_strategy\fR Solaris DDI
671 \fBmakecom_g0\fR Solaris DDI
672 \fBmakecom_g0_s\fR Solaris DDI
673 \fBmakecom_g1\fR Solaris DDI
674 \fBmakecom_g5\fR Solaris DDI
675 \fBmakedevice\fR DDI/DKI
676 \fBmax\fR DDI/DKI
677 \fBmax\fR DDI/DKI
678 \fBmambar_ops\fR Solaris DDI
679 \fBmemchr\fR Solaris DDI
680 \fBminphys\fR Solaris DDI
681 \fBmod_info\fR Solaris DDI
682 \fBmod_install\fR Solaris DDI
683 \fBmod_remove\fR Solaris DDI
684 \fBmutex_destroy\fR Solaris DDI
685 \fBmutex_enter\fR Solaris DDI
686 \fBmutex_exit\fR Solaris DDI
687 \fBmutex_init\fR Solaris DDI
688 \fBmutex_owned\fR Solaris DDI
689 \fBmutex_tryenter\fR Solaris DDI
690 \fBnochpoll\fR Solaris DDI
691 \fBnodev\fR DDI/DKI
692 \fBnulldev\fR DDI/DKI
693 \fBnumtos\fR Solaris DDI
694 \fBnvlst_add_boolean\fR Solaris DDI
695 \fBnvlst_alloc\fR Solaris DDI
696 \fBnvlst_lookup_boolean\fR Solaris DDI
697 \fBnvlst_lookup_nvpair\fR Solaris DDI
698 \fBnvlst_next_nvpair\fR Solaris DDI
699 \fBnvlst_remove\fR Solaris DDI
700 \fBnvlst_value_byte\fR Solaris DDI
701 \fBoutb\fR Solaris x86 DDI
702 \fBoutl\fR Solaris x86 DDI
703 \fBoutw\fR Solaris x86 DDI
704 \fBpci_config_getl6\fR Solaris DDI
705 \fBpci_config_get32\fR Solaris DDI
706 \fBpci_config_get64\fR Solaris DDI
707 \fBpci_config_get8\fR Solaris DDI
708 \fBpci_config_getb\fR Solaris DDI
709 \fBpci_config_getl\fR Solaris DDI
710 \fBpci_config_getw\fR Solaris DDI
711 \fBpci_config_putl6\fR Solaris DDI
712 \fBpci_config_put32\fR Solaris DDI
713 \fBpci_config_put64\fR Solaris DDI
714 \fBpci_config_put8\fR Solaris DDI
715 \fBpci_config_putb\fR Solaris DDI

```

```

716 \fBpci_config_putl\fR Solaris DDI
717 \fBpci_config_putw\fR Solaris DDI
718 \fBpci_config_setup\fR Solaris DDI
719 \fBpci_config_teardown\fR Solaris DDI
720 \fBpci_ereport_setup\fR Solaris DDI
721 \fBpci_report_pmcap\fR Solaris DDI
722 \fBpci_save_config_regs\fR Solaris DDI
723 \fBphysio\fR Solaris DDI
724 \fBpms_busy_component\fR Solaris DDI
725 \fBpms_power_has_changed\fR Solaris DDI
726 \fBpms_raise_power\fR Solaris DDI
727 \fBpms_trans_check\fR Solaris DDI
728 \fBpollwakep\fR DDI/DKI
729 \fBpci_config_teardown\fR Solaris DDI
730 \fBpci_config_teardown\fR Solaris DDI
731 \fBpriv_getbyname\fR Solaris DDI
732 \fBpriv_policy\fR Solaris DDI
733 \fBproc_signal\fR Solaris DDI
734 \fBproc_unref\fR Solaris DDI
735 \fBptob\fR DDI/DKI
736 \fBrepinsb\fR Solaris x86 DDI
737 \fBrepinsd\fR Solaris x86 DDI
738 \fBrepinsw\fR Solaris x86 DDI
739 \fBrepoutsb\fR Solaris x86 DDI
740 \fBrepoutsd\fR Solaris x86 DDI
741 \fBrepoutsw\fR Solaris x86 DDI
742 \fBrmalloc\fR DDI/DKI
743 \fBrmalloc_wait\fR DDI/DKI
744 \fBrmallocmap\fR DDI/DKI
745 \fBrmallocmap_wait\fR DDI/DKI
746 \fBrmfree\fR DDI/DKI
747 \fBrmfreemap\fR DDI/DKI
748 \fBbrw_destroy\fR Solaris DDI
749 \fBbrw_downgrade\fR Solaris DDI
750 \fBbrw_enter\fR Solaris DDI
751 \fBbrw_exit\fR Solaris DDI
752 \fBbrw_init\fR Solaris DDI
753 \fBbrw_read_locked\fR Solaris DDI
754 \fBbrw_tryenter\fR Solaris DDI
755 \fBbrw_tryupgrade\fR Solaris DDI
756 \fBscsi_abort\fR Solaris DDI
757 \fBscsi_alloc_consistent_buf\fR Solaris DDI
758 \fBscsi_cname\fR Solaris DDI
759 \fBscsi_destroy_pkt\fR Solaris DDI
760 \fBscsi_dmafree\fR Solaris DDI
761 \fBscsi_dmaget\fR Solaris DDI
762 \fBscsi_dname\fR Solaris DDI
763 \fBscsi_errmsg\fR Solaris DDI
764 \fBscsi_ext_sense_fields\fR Solaris DDI
765 \fBscsi_find_sense_descr\fR Solaris DDI
766 \fBscsi_free_consistent_buf\fR Solaris DDI
767 \fBscsi_get_device_type_scsi_options\fR Solaris DDI
768 \fBscsi_get_device_type_string\fR Solaris DDI
769 \fBscsi_hba_attach\fR Solaris DDI
770 \fBscsi_hba_attach_setup\fR Solaris DDI
771 \fBscsi_hba_detach\fR Solaris DDI
772 \fBscsi_hba_fini\fR Solaris DDI
773 \fBscsi_hba_init\fR Solaris DDI
774 \fBscsi_hba_lookup_capstr\fR Solaris DDI
775 \fBscsi_hba_pkt_alloc\fR Solaris DDI
776 \fBscsi_hba_free\fR Solaris DDI
777 \fBscsi_hba_probe\fR Solaris DDI
778 \fBscsi_hba_tran_alloc\fR Solaris DDI
779 \fBscsi_hba_tran_free\fR Solaris DDI
780 \fBscsi_ifgetcap\fR Solaris DDI

```

```

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

```

```

847 \fBusb_pipe_get_max_bulk_transfer_size\fR Solaris DDI
848 \fBusb_pipe_get_state\fR Solaris DDI
849 \fBusb_pipe_intr_xfer\fR Solaris DDI
850 \fBusb_pipe_isoc_xfer\fR Solaris DDI
851 \fBusb_pipe_open\fR Solaris DDI
852 \fBusb_pipe_reset\fR Solaris DDI
853 \fBusb_pipe_set_private\fR Solaris DDI
854 \fBusb_register_hotplug_cbs\fR Solaris DDI
855 \fBusb_reset_device\fR Solaris DDI
856 \fBuwwritec\fR DDI/DKI
857 \fBva_arg\fR Solaris DDI
858 \fBva_end\fR Solaris DDI
859 \fBva_start\fR Solaris DDI
860 \fBvcmm_err\fR DDI/DKI
861 \fBvsprintf\fR Solaris DDI
862 .TE

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

```

62774 Sat May 24 17:48:26 2014

new/usr/src/man/man9f/Makefile

4888 Undocument dma_req(9s)

4884 EOF scsi_hba_attach

4886 EOF ddi_dmae_getlim

4887 EOF ddi_iomin

4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)

4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free

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 #
11 #
12 #
13 # Copyright 2011, Richard Lowe
14 # Copyright 2012 Garrett D'Amore <garrett@damore>. All rights reserved.
15 # Copyright 2013 Nexenta Systems, Inc. All rights reserved.
16 # Copyright 2014 Garrett D'Amore <garrett@damore>

18 include \$(SRC)/Makefile.master

20 MANSECT= 9f

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

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


```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

451 scsi_reset_notify.9f //
452 scsi_sense_key.9f //
453 scsi_setup_cdb.9f //
454 scsi_slave.9f //
455 scsi_sync_pkt.9f //
456 scsi_transport.9f //
457 scsi_unprobe.9f //
458 scsi_validate_sense.9f //
459 scsi_vu_errmsg.9f //
460 semaphore.9f //
461 stoi.9f //
462 string.9f //
463 strlog.9f //
464 strqget.9f //
465 strqset.9f //
466 swab.9f //
467 taskq.9f //
468 testb.9f //
469 timeout.9f //
470 u8_strcmp.9f //
471 u8_textprep_str.9f //
472 u8_validate.9f //
473 uconv_ul6tou32.9f //
474 uimove.9f //
475 unbufcall.9f //
476 unlinkb.9f //
477 untimeout.9f //
478 ureadc.9f //
479 usb_alloc_request.9f //
480 usb_client_attach.9f //
481 usb_clr_feature.9f //
482 usb_create_pm_components.9f //
483 usb_get_addr.9f //
484 usb_get_alt_if.9f //
485 usb_get_cfg.9f //
486 usb_get_current_frame_number.9f //
487 usb_get_dev_data.9f //
488 usb_get_max_pkts_per_isoc_request.9f //
489 usb_get_status.9f //
490 usb_get_string_descr.9f //
491 usb_handle_remote_wakeup.9f //
492 usb_lookup_ep_data.9f //
493 usb_parse_data.9f //
494 usb_pipe_bulk_xfer.9f //
495 usb_pipe_close.9f //
496 usb_pipe_ctrl_xfer.9f //
497 usb_pipe_drain_reqs.9f //
498 usb_pipe_get_max_bulk_transfer_size.9f //
499 usb_pipe_get_state.9f //
500 usb_pipe_intr_xfer.9f //
501 usb_pipe_isoc_xfer.9f //
502 usb_pipe_open.9f //
503 usb_pipe_reset.9f //
504 usb_pipe_set_private.9f //
505 usb_register_hotplug_cbs.9f //
506 usb_reset_device.9f //
507 uwritec.9f //
508 va_arg.9f //
509 vsprintf.9f //

511 MANLINKS= //
512 SIZEOF_PTR.9f //
513 SIZEOF_STRUCT.9f //
514 STRUCT_BUF.9f //
515 STRUCT_FADDR.9f //
516 STRUCT_FGET.9f //
517 STRUCT_FGETP.9f //

```

```

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

```

```

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

```

```

649 csx_Parse_CISTPL_LONGLINK_C.9f  \
650 csx_Put16.9f                    \
651 csx_Put32.9f                    \
652 csx_Put64.9f                    \
653 csx_ReleaseIO.9f                \
654 csx_ReleaseIRQ.9f               \
655 csx_ReleaseSocketMask.9f        \
656 csx_ReleaseWindow.9f            \
657 csx_RemoveDeviceNode.9f         \
658 csx_RepGet16.9f                  \
659 csx_RepGet32.9f                  \
660 csx_RepGet64.9f                  \
661 csx_RepPut16.9f                  \
662 csx_RepPut32.9f                  \
663 csx_RepPut64.9f                  \
664 cv_broadcast.9f                  \
665 cv_destroy.9f                    \
666 cv_init.9f                       \
667 cv_reltimedwait.9f               \
668 cv_reltimedwait_sig.9f          \
669 cv_signal.9f                     \
670 cv_timedwait.9f                 \
671 cv_timedwait_sig.9f             \
672 cv_wait.9f                       \
673 cv_wait_sig.9f                  \
674 ddi_btopr.9f                     \
675 ddi_cb_unregister.9f             \
676 ddi_check_dma_handle.9f         \
677 ddi_devid_free.9f                \
678 ddi_devid_get.9f                 \
679 ddi_devid_init.9f                \
680 ddi_devid_register.9f            \
681 ddi_devid_sizeof.9f              \
682 ddi_devid_str_decode.9f          \
683 ddi_devid_str_encode.9f          \
684 ddi_devid_str_free.9f            \
685 ddi_devid_unregister.9f          \
686 ddi_devid_valid.9f               \
687 ddi_devmap_segmap.9f             \
688 ddi_dmae_lstparty.9f             \
689 ddi_dmae_alloc.9f                \
690 ddi_dmae_disable.9f              \
691 ddi_dmae_enable.9f               \
692 ddi_dmae_getattr.9f              \
693 ddi_dmae_getcnt.9f               \
695 ddi_dmae_getlim.9f               \
694 ddi_dmae_prog.9f                 \
695 ddi_dmae_release.9f              \
696 ddi_dmae_stop.9f                 \
697 ddi_exit_critical.9f             \
698 ddi_fls.9f                       \
699 ddi_fm_capable.9f                \
700 ddi_fm_dma_err_clear.9f          \
701 ddi_fm_dma_err_get.9f            \
702 ddi_fm_fini.9f                   \
703 ddi_fm_handler_unregister.9f     \
704 ddi_get16.9f                     \
705 ddi_get32.9f                     \
706 ddi_get64.9f                     \
707 ddi_get_iblock_cookie.9f         \
708 ddi_get_lbolt64.9f               \
709 ddi_get_name.9f                  \
710 ddi_get_soft_iblock_cookie.9f    \
711 ddi_get_soft_state.9f            \
712 ddi_getb.9f                      \
713 ddi_getl.9f                      \

```

```

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

```

```

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

```

```

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

```

```

912 list_insert_before.9f //
913 list_insert_head.9f //
914 list_insert_tail.9f //
915 list_is_empty.9f //
916 list_link_active.9f //
917 list_link_init.9f //
918 list_link_replace.9f //
919 list_move_tail.9f //
920 list_next.9f //
921 list_prev.9f //
922 list_remove.9f //
923 list_remove_head.9f //
924 list_remove_tail.9f //
925 list_tail.9f //
926 makecom_g0.9f //
927 makecom_g0_s.9f //
928 makecom_g1.9f //
929 makecom_g5.9f //
930 membar_consumer.9f //
931 membar_enter.9f //
932 membar_exit.9f //
933 membar_producer.9f //
934 memcmp.9f //
935 memcpy.9f //
936 memmove.9f //
937 memset.9f //
938 minphys.9f //
939 mod_info.9f //
940 mod_modname.9f //
941 mod_remove.9f //
942 mutex_destroy.9f //
943 mutex_enter.9f //
944 mutex_exit.9f //
945 mutex_init.9f //
946 mutex_owned.9f //
947 mutex_tryenter.9f //
948 net_event_notify_unregister.9f //
949 net_instance_notify_unregister.9f //
950 net_instance_protocol_unregister.9f //
951 numtos.9f //
952 nv_alloc_fini.9f //
953 nv_alloc_init.9f //
954 nvlist_add_boolean_array.9f //
955 nvlist_add_boolean_value.9f //
956 nvlist_add_byte.9f //
957 nvlist_add_byte_array.9f //
958 nvlist_add_int16.9f //
959 nvlist_add_int16_array.9f //
960 nvlist_add_int32.9f //
961 nvlist_add_int32_array.9f //
962 nvlist_add_int64.9f //
963 nvlist_add_int64_array.9f //
964 nvlist_add_int8.9f //
965 nvlist_add_int8_array.9f //
966 nvlist_add_nvlist.9f //
967 nvlist_add_nvlist_array.9f //
968 nvlist_add_nvpair.9f //
969 nvlist_add_string.9f //
970 nvlist_add_string_array.9f //
971 nvlist_add_uint16.9f //
972 nvlist_add_uint16_array.9f //
973 nvlist_add_uint32.9f //
974 nvlist_add_uint32_array.9f //
975 nvlist_add_uint64.9f //
976 nvlist_add_uint64_array.9f //
977 nvlist_add_uint8.9f //

```

```

978 nvlist_add_uint8_array.9f //
979 nvlist_dup.9f //
980 nvlist_exists.9f //
981 nvlist_free.9f //
982 nvlist_lookup_boolean_array.9f //
983 nvlist_lookup_boolean_value.9f //
984 nvlist_lookup_byte.9f //
985 nvlist_lookup_byte_array.9f //
986 nvlist_lookup_int16.9f //
987 nvlist_lookup_int16_array.9f //
988 nvlist_lookup_int32.9f //
989 nvlist_lookup_int32_array.9f //
990 nvlist_lookup_int64.9f //
991 nvlist_lookup_int64_array.9f //
992 nvlist_lookup_int8.9f //
993 nvlist_lookup_int8_array.9f //
994 nvlist_lookup_nvlist.9f //
995 nvlist_lookup_nvlist_array.9f //
996 nvlist_lookup_pairs.9f //
997 nvlist_lookup_string.9f //
998 nvlist_lookup_string_array.9f //
999 nvlist_lookup_uint16.9f //
1000 nvlist_lookup_uint16_array.9f //
1001 nvlist_lookup_uint32.9f //
1002 nvlist_lookup_uint32_array.9f //
1003 nvlist_lookup_uint64.9f //
1004 nvlist_lookup_uint64_array.9f //
1005 nvlist_lookup_uint8.9f //
1006 nvlist_lookup_uint8_array.9f //
1007 nvlist_merge.9f //
1008 nvlist_pack.9f //
1009 nvlist_remove_all.9f //
1010 nvlist_size.9f //
1011 nvlist_t.9f //
1012 nvlist_unpack.9f //
1013 nvlist_xalloc.9f //
1014 nvlist_xdup.9f //
1015 nvlist_xpack.9f //
1016 nvlist_xunpack.9f //
1017 nvpair_name.9f //
1018 nvpair_type.9f //
1019 nvpair_value_boolean_array.9f //
1020 nvpair_value_byte_array.9f //
1021 nvpair_value_int16.9f //
1022 nvpair_value_int16_array.9f //
1023 nvpair_value_int32.9f //
1024 nvpair_value_int32_array.9f //
1025 nvpair_value_int64.9f //
1026 nvpair_value_int64_array.9f //
1027 nvpair_value_int8.9f //
1028 nvpair_value_int8_array.9f //
1029 nvpair_value_nvlist.9f //
1030 nvpair_value_nvlist_array.9f //
1031 nvpair_value_string.9f //
1032 nvpair_value_string_array.9f //
1033 nvpair_value_uint16.9f //
1034 nvpair_value_uint16_array.9f //
1035 nvpair_value_uint32.9f //
1036 nvpair_value_uint32_array.9f //
1037 nvpair_value_uint64.9f //
1038 nvpair_value_uint64_array.9f //
1039 nvpair_value_uint8.9f //
1040 nvpair_value_uint8_array.9f //
1041 otherq.9f //
1042 outl.9f //
1043 outw.9f //

```

```

1044 pci_config_get16.9f //
1045 pci_config_get32.9f //
1046 pci_config_get64.9f //
1047 pci_config_getb.9f //
1048 pci_config_getl.9f //
1049 pci_config_getll.9f //
1050 pci_config_getw.9f //
1051 pci_config_put16.9f //
1052 pci_config_put32.9f //
1053 pci_config_put64.9f //
1054 pci_config_put8.9f //
1055 pci_config_putb.9f //
1056 pci_config_putl.9f //
1057 pci_config_putll.9f //
1058 pci_config_putw.9f //
1059 pci_config_teardown.9f //
1060 pci_ereport_post.9f //
1061 pci_ereport_teardown.9f //
1062 pci_restore_config_regs.9f //
1063 pm_idle_component.9f //
1064 pm_lower_power.9f //
1065 priv_policy_choice.9f //
1066 priv_policy_only.9f //
1067 proc_ref.9f //
1068 proc_unref.9f //
1069 qprocsoff.9f //
1070 qwait_sig.9f //
1071 rd.9f //
1072 repinsb.9f //
1073 repinsd.9f //
1074 repinsw.9f //
1075 repoutsb.9f //
1076 repoutsd.9f //
1077 repoutsw.9f //
1078 rmallocmap_wait.9f //
1079 rmfreemap.9f //
1080 rw_destroy.9f //
1081 rw_downgrade.9f //
1082 rw_enter.9f //
1083 rw_exit.9f //
1084 rw_init.9f //
1085 rw_read_locked.9f //
1086 rw_tryenter.9f //
1087 rw_tryupgrade.9f //
1088 samestr.9f //
1089 scsi_dmafree.9f //
1090 scsi_dname.9f //
1093 scsi_hba_attach.9f //
1091 scsi_hba_detach.9f //
1092 scsi_hba_fini.9f //
1093 scsi_hba_pkt_free.9f //
1094 scsi_hba_tran_free.9f //
1095 scsi_ifsetcap.9f //
1096 scsi_mname.9f //
1097 scsi_pktfree.9f //
1098 scsi_resalloc.9f //
1099 scsi_resfree.9f //
1100 scsi_rname.9f //
1101 scsi_sense_asc.9f //
1102 scsi_sense_ascq.9f //
1103 scsi_sense_cmdspecific_uint64.9f //
1104 scsi_sense_info_uint64.9f //
1105 scsi_sname.9f //
1106 scsi_unslave.9f //
1107 sema_destroy.9f //
1108 sema_init.9f //

```

```

1109 sema_p.9f //
1110 sema_p_sig.9f //
1111 sema_tryp.9f //
1112 sema_v.9f //
1113 strcasecmp.9f //
1114 strchr.9f //
1115 strcmp.9f //
1116 strcpy.9f //
1117 strdup.9f //
1118 strfree.9f //
1119 strlcat.9f //
1120 strlcpy.9f //
1121 strlen.9f //
1122 strncasecmp.9f //
1123 strncat.9f //
1124 strncmp.9f //
1125 strncpy.9f //
1126 strnlen.9f //
1127 strrchr.9f //
1128 strspn.9f //
1129 taskq_suspended.9f //
1130 uconv_u16tou8.9f //
1131 uconv_u32tou16.9f //
1132 uconv_u32tou8.9f //
1133 uconv_u8tou16.9f //
1134 uconv_u8tou32.9f //
1135 unfreeze_str.9f //
1136 usb_alloc_bulk_req.9f //
1137 usb_alloc_ctrl_req.9f //
1138 usb_alloc_intr_req.9f //
1139 usb_alloc_isoc_req.9f //
1140 usb_client_detach.9f //
1141 usb_free_bulk_req.9f //
1142 usb_free_ctrl_req.9f //
1143 usb_free_descr_tree.9f //
1144 usb_free_dev_data.9f //
1145 usb_free_intr_req.9f //
1146 usb_free_isoc_req.9f //
1147 usb_get_if_number.9f //
1148 usb_owns_device.9f //
1149 usb_pipe_ctrl_xfer_wait.9f //
1150 usb_pipe_get_private.9f //
1151 usb_pipe_stop_intr_polling.9f //
1152 usb_pipe_stop_isoc_polling.9f //
1153 usb_print_descr_tree.9f //
1154 usb_set_alt_if.9f //
1155 usb_set_cfg.9f //
1156 usb_unregister_hotplug_cbs.9f //
1157 va_copy.9f //
1158 va_end.9f //
1159 va_start.9f //
1160 vcmn_err.9f //
1161 wr.9f //
1162 zcmn_err.9f //

1164 assert.9f := LINKSRC = ASSERT.9f
1166 intro.9f := LINKSRC = Intro.9f
1168 otherq.9f := LINKSRC = OTHERQ.9f
1170 rd.9f := LINKSRC = RD.9f
1172 samestr.9f := LINKSRC = SAMESTR.9f
1174 SIZEOF_PTR.9f := LINKSRC = STRUCT_DECL.9f

```



```

1175 SIZEOF_STRUCT.9f      := LINKSRC = STRUCT_DECL.9f
1176 STRUCT_BUF.9f         := LINKSRC = STRUCT_DECL.9f
1177 STRUCT_FADDR.9f       := LINKSRC = STRUCT_DECL.9f
1178 STRUCT_FGET.9f        := LINKSRC = STRUCT_DECL.9f
1179 STRUCT_FGETP.9f       := LINKSRC = STRUCT_DECL.9f
1180 STRUCT_FSET.9f        := LINKSRC = STRUCT_DECL.9f
1181 STRUCT_FSETP.9f       := LINKSRC = STRUCT_DECL.9f
1182 STRUCT_HANDLE.9f      := LINKSRC = STRUCT_DECL.9f
1183 STRUCT_INIT.9f        := LINKSRC = STRUCT_DECL.9f
1184 STRUCT_SET_HANDLE.9f  := LINKSRC = STRUCT_DECL.9f
1185 STRUCT_SIZE.9f        := LINKSRC = STRUCT_DECL.9f

1187 wr.9f                 := LINKSRC = WR.9f

1189 atomic_add_16.9f      := LINKSRC = atomic_add.9f
1190 atomic_add_16_nv.9f   := LINKSRC = atomic_add.9f
1191 atomic_add_32.9f      := LINKSRC = atomic_add.9f
1192 atomic_add_32_nv.9f   := LINKSRC = atomic_add.9f
1193 atomic_add_64.9f      := LINKSRC = atomic_add.9f
1194 atomic_add_64_nv.9f   := LINKSRC = atomic_add.9f
1195 atomic_add_8.9f       := LINKSRC = atomic_add.9f
1196 atomic_add_8_nv.9f   := LINKSRC = atomic_add.9f
1197 atomic_add_char.9f   := LINKSRC = atomic_add.9f
1198 atomic_add_char_nv.9f := LINKSRC = atomic_add.9f
1199 atomic_add_int.9f     := LINKSRC = atomic_add.9f
1200 atomic_add_int_nv.9f  := LINKSRC = atomic_add.9f
1201 atomic_add_long.9f    := LINKSRC = atomic_add.9f
1202 atomic_add_long_nv.9f := LINKSRC = atomic_add.9f
1203 atomic_add_ptr.9f     := LINKSRC = atomic_add.9f
1204 atomic_add_ptr_nv.9f  := LINKSRC = atomic_add.9f
1205 atomic_add_short.9f   := LINKSRC = atomic_add.9f
1206 atomic_add_short_nv.9f := LINKSRC = atomic_add.9f
1207 atomic_and_16.9f     := LINKSRC = atomic_and.9f
1208 atomic_and_16_nv.9f  := LINKSRC = atomic_and.9f
1209 atomic_and_32.9f     := LINKSRC = atomic_and.9f
1210 atomic_and_32_nv.9f  := LINKSRC = atomic_and.9f
1211 atomic_and_64.9f     := LINKSRC = atomic_and.9f
1212 atomic_and_64_nv.9f  := LINKSRC = atomic_and.9f
1213 atomic_and_8.9f      := LINKSRC = atomic_and.9f
1214 atomic_and_8_nv.9f   := LINKSRC = atomic_and.9f
1215 atomic_and_uchar.9f  := LINKSRC = atomic_and.9f
1216 atomic_and_uchar_nv.9f := LINKSRC = atomic_and.9f
1217 atomic_and_uint.9f   := LINKSRC = atomic_and.9f
1218 atomic_and_uint_nv.9f := LINKSRC = atomic_and.9f
1219 atomic_and_ulong.9f  := LINKSRC = atomic_and.9f
1220 atomic_and_ulong_nv.9f := LINKSRC = atomic_and.9f
1221 atomic_and_ushort.9f := LINKSRC = atomic_and.9f
1222 atomic_and_ushort_nv.9f := LINKSRC = atomic_and.9f

1224 atomic_clear_long_excl.9f := LINKSRC = atomic_bits.9f
1225 atomic_set_long_excl.9f  := LINKSRC = atomic_bits.9f

1227 atomic_cas_16.9f     := LINKSRC = atomic_cas.9f
1228 atomic_cas_32.9f     := LINKSRC = atomic_cas.9f
1229 atomic_cas_64.9f     := LINKSRC = atomic_cas.9f
1230 atomic_cas_8.9f      := LINKSRC = atomic_cas.9f
1231 atomic_cas_ptr.9f    := LINKSRC = atomic_cas.9f
1232 atomic_cas_uchar.9f  := LINKSRC = atomic_cas.9f
1233 atomic_cas_uint.9f   := LINKSRC = atomic_cas.9f
1234 atomic_cas_ulong.9f  := LINKSRC = atomic_cas.9f
1235 atomic_cas_ushort.9f := LINKSRC = atomic_cas.9f

1237 atomic_dec_16.9f     := LINKSRC = atomic_dec.9f
1238 atomic_dec_16_nv.9f  := LINKSRC = atomic_dec.9f
1239 atomic_dec_32.9f     := LINKSRC = atomic_dec.9f
1240 atomic_dec_32_nv.9f  := LINKSRC = atomic_dec.9f

```

```

1241 atomic_dec_64.9f     := LINKSRC = atomic_dec.9f
1242 atomic_dec_64_nv.9f  := LINKSRC = atomic_dec.9f
1243 atomic_dec_8.9f      := LINKSRC = atomic_dec.9f
1244 atomic_dec_8_nv.9f   := LINKSRC = atomic_dec.9f
1245 atomic_dec_ptr.9f    := LINKSRC = atomic_dec.9f
1246 atomic_dec_ptr_nv.9f := LINKSRC = atomic_dec.9f
1247 atomic_dec_uchar.9f := LINKSRC = atomic_dec.9f
1248 atomic_dec_uchar_nv.9f := LINKSRC = atomic_dec.9f
1249 atomic_dec_uint.9f   := LINKSRC = atomic_dec.9f
1250 atomic_dec_uint_nv.9f := LINKSRC = atomic_dec.9f
1251 atomic_dec_ulong.9f  := LINKSRC = atomic_dec.9f
1252 atomic_dec_ulong_nv.9f := LINKSRC = atomic_dec.9f
1253 atomic_dec_ushort.9f := LINKSRC = atomic_dec.9f
1254 atomic_dec_ushort_nv.9f := LINKSRC = atomic_dec.9f

1256 atomic_inc_16.9f     := LINKSRC = atomic_inc.9f
1257 atomic_inc_16_nv.9f  := LINKSRC = atomic_inc.9f
1258 atomic_inc_32.9f     := LINKSRC = atomic_inc.9f
1259 atomic_inc_32_nv.9f  := LINKSRC = atomic_inc.9f
1260 atomic_inc_64.9f     := LINKSRC = atomic_inc.9f
1261 atomic_inc_64_nv.9f  := LINKSRC = atomic_inc.9f
1262 atomic_inc_8.9f      := LINKSRC = atomic_inc.9f
1263 atomic_inc_8_nv.9f   := LINKSRC = atomic_inc.9f
1264 atomic_inc_ptr.9f    := LINKSRC = atomic_inc.9f
1265 atomic_inc_ptr_nv.9f := LINKSRC = atomic_inc.9f
1266 atomic_inc_uchar.9f := LINKSRC = atomic_inc.9f
1267 atomic_inc_uchar_nv.9f := LINKSRC = atomic_inc.9f
1268 atomic_inc_uint.9f   := LINKSRC = atomic_inc.9f
1269 atomic_inc_uint_nv.9f := LINKSRC = atomic_inc.9f
1270 atomic_inc_ulong.9f  := LINKSRC = atomic_inc.9f
1271 atomic_inc_ulong_nv.9f := LINKSRC = atomic_inc.9f
1272 atomic_inc_ushort.9f := LINKSRC = atomic_inc.9f
1273 atomic_inc_ushort_nv.9f := LINKSRC = atomic_inc.9f

1275 atomic_or_16.9f      := LINKSRC = atomic_or.9f
1276 atomic_or_16_nv.9f  := LINKSRC = atomic_or.9f
1277 atomic_or_32.9f     := LINKSRC = atomic_or.9f
1278 atomic_or_32_nv.9f  := LINKSRC = atomic_or.9f
1279 atomic_or_64.9f     := LINKSRC = atomic_or.9f
1280 atomic_or_64_nv.9f  := LINKSRC = atomic_or.9f
1281 atomic_or_8.9f       := LINKSRC = atomic_or.9f
1282 atomic_or_8_nv.9f   := LINKSRC = atomic_or.9f
1283 atomic_or_uchar.9f  := LINKSRC = atomic_or.9f
1284 atomic_or_uchar_nv.9f := LINKSRC = atomic_or.9f
1285 atomic_or_uint.9f   := LINKSRC = atomic_or.9f
1286 atomic_or_uint_nv.9f := LINKSRC = atomic_or.9f
1287 atomic_or_ulong.9f  := LINKSRC = atomic_or.9f
1288 atomic_or_ulong_nv.9f := LINKSRC = atomic_or.9f
1289 atomic_or_ushort.9f := LINKSRC = atomic_or.9f
1290 atomic_or_ushort_nv.9f := LINKSRC = atomic_or.9f

1292 atomic_swap_16.9f    := LINKSRC = atomic_swap.9f
1293 atomic_swap_32.9f    := LINKSRC = atomic_swap.9f
1294 atomic_swap_64.9f    := LINKSRC = atomic_swap.9f
1295 atomic_swap_8.9f     := LINKSRC = atomic_swap.9f
1296 atomic_swap_ptr.9f   := LINKSRC = atomic_swap.9f
1297 atomic_swap_uchar.9f := LINKSRC = atomic_swap.9f
1298 atomic_swap_uint.9f  := LINKSRC = atomic_swap.9f
1299 atomic_swap_ulong.9f := LINKSRC = atomic_swap.9f
1300 atomic_swap_ushort.9f := LINKSRC = atomic_swap.9f

1302 vcmn_err.9f          := LINKSRC = cmn_err.9f
1303 zcmn_err.9f          := LINKSRC = cmn_err.9f

1305 cv_broadcast.9f     := LINKSRC = condvar.9f
1306 cv_destroy.9f        := LINKSRC = condvar.9f

```

```

1307 cv_init.9f           := LINKSRC = condvar.9f
1308 cv_reltimedwait.9f   := LINKSRC = condvar.9f
1309 cv_reltimedwait_sig.9f := LINKSRC = condvar.9f
1310 cv_signal.9f         := LINKSRC = condvar.9f
1311 cv_timedwait.9f      := LINKSRC = condvar.9f
1312 cv_timedwait_sig.9f  := LINKSRC = condvar.9f
1313 cv_wait.9f          := LINKSRC = condvar.9f
1314 cv_wait_sig.9f       := LINKSRC = condvar.9f

1316 csx_Get16.9f         := LINKSRC = csx_Get8.9f
1317 csx_Get32.9f         := LINKSRC = csx_Get8.9f
1318 csx_Get64.9f         := LINKSRC = csx_Get8.9f

1320 csx_GetNextClient.9f := LINKSRC = csx_GetFirstClient.9f

1322 csx_GetNextTuple.9f := LINKSRC = csx_GetFirstTuple.9f

1324 csx_RemoveDeviceNode.9f := LINKSRC = csx_MakeDeviceNode.9f

1326 csx_Parse_CISTPL_DEVICE_A.9f := LINKSRC = csx_Parse_CISTPL_DEVICE.9f
1327 csx_Parse_CISTPL_DEVICE_OA.9f := LINKSRC = csx_Parse_CISTPL_DEVICE.9f
1328 csx_Parse_CISTPL_DEVICE_OC.9f := LINKSRC = csx_Parse_CISTPL_DEVICE.9f

1330 csx_Parse_CISTPL_JEDEC_A.9f := LINKSRC = csx_Parse_CISTPL_JEDEC_C.9f

1332 csx_Parse_CISTPL_LONGLINK_C.9f := LINKSRC = csx_Parse_CISTPL_LONGLINK_A

1334 csx_Put16.9f         := LINKSRC = csx_Put8.9f
1335 csx_Put32.9f         := LINKSRC = csx_Put8.9f
1336 csx_Put64.9f         := LINKSRC = csx_Put8.9f

1338 csx_RepGet16.9f      := LINKSRC = csx_RepGet8.9f
1339 csx_RepGet32.9f     := LINKSRC = csx_RepGet8.9f
1340 csx_RepGet64.9f     := LINKSRC = csx_RepGet8.9f

1342 csx_RepPut16.9f     := LINKSRC = csx_RepPut8.9f
1343 csx_RepPut32.9f     := LINKSRC = csx_RepPut8.9f
1344 csx_RepPut64.9f     := LINKSRC = csx_RepPut8.9f

1346 csx_ReleaseIO.9f    := LINKSRC = csx_RequestIO.9f

1348 csx_ReleaseIRQ.9f   := LINKSRC = csx_RequestIRQ.9f

1350 csx_ReleaseSocketMask.9f := LINKSRC = csx_RequestSocketMask.9f

1352 csx_ReleaseWindow.9f := LINKSRC = csx_RequestWindow.9f

1354 csx_GetEventMask.9f := LINKSRC = csx_SetEventMask.9f

1356 ddi_get_iblock_cookie.9f := LINKSRC = ddi_add_intr.9f
1357 ddi_remove_intr.9f      := LINKSRC = ddi_add_intr.9f

1359 ddi_get_soft_iblock_cookie.9f := LINKSRC = ddi_add_softintr.9f
1360 ddi_remove_softintr.9f      := LINKSRC = ddi_add_softintr.9f
1361 ddi_trigger_softintr.9f     := LINKSRC = ddi_add_softintr.9f

1363 ddi_get_name.9f       := LINKSRC = ddi_binding_name.9f

1365 ddi_btopr.9f          := LINKSRC = ddi_btop.9f
1366 ddi_ptob.9f           := LINKSRC = ddi_btop.9f

1368 ddi_cb_unregister.9f  := LINKSRC = ddi_cb_register.9f

1370 ddi_check_dma_handle.9f := LINKSRC = ddi_check_acc_handle.9f

1372 crgetgid.9f          := LINKSRC = ddi_cred.9f

```

```

1373 crgetgroups.9f       := LINKSRC = ddi_cred.9f
1374 crgetngroups.9f     := LINKSRC = ddi_cred.9f
1375 crgetrgid.9f        := LINKSRC = ddi_cred.9f
1376 crgetruid.9f        := LINKSRC = ddi_cred.9f
1377 crgetsgid.9f        := LINKSRC = ddi_cred.9f
1378 crgetsuid.9f         := LINKSRC = ddi_cred.9f
1379 crgetuid.9f         := LINKSRC = ddi_cred.9f
1380 crgetzoneid.9f       := LINKSRC = ddi_cred.9f

1382 ddi_devid_free.9f    := LINKSRC = ddi_devid_compare.9f
1383 ddi_devid_get.9f     := LINKSRC = ddi_devid_compare.9f
1384 ddi_devid_init.9f    := LINKSRC = ddi_devid_compare.9f
1385 ddi_devid_register.9f := LINKSRC = ddi_devid_compare.9f
1386 ddi_devid_sizeof.9f  := LINKSRC = ddi_devid_compare.9f
1387 ddi_devid_str_decode.9f := LINKSRC = ddi_devid_compare.9f
1388 ddi_devid_str_encode.9f := LINKSRC = ddi_devid_compare.9f
1389 ddi_devid_str_free.9f := LINKSRC = ddi_devid_compare.9f
1390 ddi_devid_unregister.9f := LINKSRC = ddi_devid_compare.9f
1391 ddi_devid_valid.9f    := LINKSRC = ddi_devid_compare.9f

1393 ddi_dmae_lstparty.9f := LINKSRC = ddi_dmae.9f
1394 ddi_dmae_alloc.9f   := LINKSRC = ddi_dmae.9f
1395 ddi_dmae_disable.9f := LINKSRC = ddi_dmae.9f
1396 ddi_dmae_enable.9f  := LINKSRC = ddi_dmae.9f
1397 ddi_dmae_getattr.9f := LINKSRC = ddi_dmae.9f
1398 ddi_dmae_getcnt.9f  := LINKSRC = ddi_dmae.9f
1402 ddi_dmae_getlim.9f := LINKSRC = ddi_dmae.9f
1399 ddi_dmae_prog.9f   := LINKSRC = ddi_dmae.9f
1400 ddi_dmae_release.9f := LINKSRC = ddi_dmae.9f
1401 ddi_dmae_stop.9f    := LINKSRC = ddi_dmae.9f

1403 ddi_exit_critical.9f := LINKSRC = ddi_enter_critical.9f

1405 ddi_fls.9f           := LINKSRC = ddi_ffs.9f

1407 ddi_fm_dma_err_clear.9f := LINKSRC = ddi_fm_acc_err_clear.9f

1409 ddi_fm_dma_err_get.9f  := LINKSRC = ddi_fm_acc_err_get.9f

1411 ddi_fm_handler_unregister.9f := LINKSRC = ddi_fm_handler_register.9f

1413 ddi_fm_capable.9f      := LINKSRC = ddi_fm_init.9f
1414 ddi_fm_fini.9f        := LINKSRC = ddi_fm_init.9f

1416 ddi_get16.9f          := LINKSRC = ddi_get8.9f
1417 ddi_get32.9f          := LINKSRC = ddi_get8.9f
1418 ddi_get64.9f          := LINKSRC = ddi_get8.9f
1419 ddi_getb.9f           := LINKSRC = ddi_get8.9f
1420 ddi_getl.9f           := LINKSRC = ddi_get8.9f
1421 ddi_getll.9f          := LINKSRC = ddi_get8.9f
1422 ddi_getw.9f           := LINKSRC = ddi_get8.9f

1424 ddi_set_driver_private.9f := LINKSRC = ddi_get_driver_private.9f

1426 ddi_get_lbolt64.9f    := LINKSRC = ddi_get_lbolt.9f

1428 ddi_intr_remove_handler.9f := LINKSRC = ddi_intr_add_handler.9f

1430 ddi_intr_get_softint_pri.9f := LINKSRC = ddi_intr_add_softint.9f
1431 ddi_intr_remove_softint.9f := LINKSRC = ddi_intr_add_softint.9f
1432 ddi_intr_set_softint_pri.9f := LINKSRC = ddi_intr_add_softint.9f
1433 ddi_intr_trigger_softint.9f := LINKSRC = ddi_intr_add_softint.9f

1435 ddi_intr_free.9f       := LINKSRC = ddi_intr_alloc.9f

1437 ddi_intr_block_disable.9f := LINKSRC = ddi_intr_enable.9f

```

```

1438 ddi_intr_block_enable.9f      := LINKSRC = ddi_intr_enable.9f
1439 ddi_intr_disable.9f          := LINKSRC = ddi_intr_enable.9f

1441 ddi_intr_set_cap.9f          := LINKSRC = ddi_intr_get_cap.9f

1443 ddi_intr_get_navail.9f       := LINKSRC = ddi_intr_get_nintrs.9f

1445 ddi_intr_set_pri.9f         := LINKSRC = ddi_intr_get_pri.9f

1447 ddi_intr_clr_mask.9f       := LINKSRC = ddi_intr_set_mask.9f

1449 ddi_io_get16.9f              := LINKSRC = ddi_io_get8.9f
1450 ddi_io_get32.9f              := LINKSRC = ddi_io_get8.9f
1451 ddi_io_getb.9f               := LINKSRC = ddi_io_get8.9f
1452 ddi_io_getl.9f               := LINKSRC = ddi_io_get8.9f
1453 ddi_io_getw.9f               := LINKSRC = ddi_io_get8.9f

1455 ddi_io_put16.9f              := LINKSRC = ddi_io_put8.9f
1456 ddi_io_put32.9f              := LINKSRC = ddi_io_put8.9f
1457 ddi_io_putb.9f               := LINKSRC = ddi_io_put8.9f
1458 ddi_io_putl.9f               := LINKSRC = ddi_io_put8.9f
1459 ddi_io_putw.9f               := LINKSRC = ddi_io_put8.9f

1461 ddi_io_rep_get16.9f          := LINKSRC = ddi_io_rep_get8.9f
1462 ddi_io_rep_get32.9f          := LINKSRC = ddi_io_rep_get8.9f
1463 ddi_io_rep_getb.9f           := LINKSRC = ddi_io_rep_get8.9f
1464 ddi_io_rep_getl.9f           := LINKSRC = ddi_io_rep_get8.9f
1465 ddi_io_rep_getw.9f           := LINKSRC = ddi_io_rep_get8.9f

1467 ddi_io_rep_put16.9f          := LINKSRC = ddi_io_rep_put8.9f
1468 ddi_io_rep_put32.9f          := LINKSRC = ddi_io_rep_put8.9f
1469 ddi_io_rep_putb.9f           := LINKSRC = ddi_io_rep_put8.9f
1470 ddi_io_rep_putl.9f           := LINKSRC = ddi_io_rep_put8.9f
1471 ddi_io_rep_putw.9f           := LINKSRC = ddi_io_rep_put8.9f

1473 ddi_unmap_regs.9f           := LINKSRC = ddi_map_regs.9f

1475 ddi_mem_get16.9f             := LINKSRC = ddi_mem_get8.9f
1476 ddi_mem_get32.9f             := LINKSRC = ddi_mem_get8.9f
1477 ddi_mem_get64.9f             := LINKSRC = ddi_mem_get8.9f
1478 ddi_mem_getb.9f              := LINKSRC = ddi_mem_get8.9f
1479 ddi_mem_getl.9f              := LINKSRC = ddi_mem_get8.9f
1480 ddi_mem_getll.9f             := LINKSRC = ddi_mem_get8.9f
1481 ddi_mem_getw.9f              := LINKSRC = ddi_mem_get8.9f

1483 ddi_mem_put16.9f             := LINKSRC = ddi_mem_put8.9f
1484 ddi_mem_put32.9f             := LINKSRC = ddi_mem_put8.9f
1485 ddi_mem_put64.9f             := LINKSRC = ddi_mem_put8.9f
1486 ddi_mem_putb.9f              := LINKSRC = ddi_mem_put8.9f
1487 ddi_mem_putl.9f              := LINKSRC = ddi_mem_put8.9f
1488 ddi_mem_putll.9f             := LINKSRC = ddi_mem_put8.9f
1489 ddi_mem_putw.9f              := LINKSRC = ddi_mem_put8.9f

1491 ddi_mem_rep_get16.9f         := LINKSRC = ddi_mem_rep_get8.9f
1492 ddi_mem_rep_get32.9f         := LINKSRC = ddi_mem_rep_get8.9f
1493 ddi_mem_rep_get64.9f         := LINKSRC = ddi_mem_rep_get8.9f
1494 ddi_mem_rep_getb.9f          := LINKSRC = ddi_mem_rep_get8.9f
1495 ddi_mem_rep_getl.9f          := LINKSRC = ddi_mem_rep_get8.9f
1496 ddi_mem_rep_getll.9f         := LINKSRC = ddi_mem_rep_get8.9f
1497 ddi_mem_rep_getw.9f          := LINKSRC = ddi_mem_rep_get8.9f

1499 ddi_mem_rep_put16.9f         := LINKSRC = ddi_mem_rep_put8.9f
1500 ddi_mem_rep_put32.9f         := LINKSRC = ddi_mem_rep_put8.9f
1501 ddi_mem_rep_put64.9f         := LINKSRC = ddi_mem_rep_put8.9f
1502 ddi_mem_rep_putb.9f          := LINKSRC = ddi_mem_rep_put8.9f
1503 ddi_mem_rep_putl.9f          := LINKSRC = ddi_mem_rep_put8.9f

```

```

1504 ddi_mem_rep_putll.9f         := LINKSRC = ddi_mem_rep_put8.9f
1505 ddi_mem_rep_putw.9f          := LINKSRC = ddi_mem_rep_put8.9f

1507 ddi_modclose.9f             := LINKSRC = ddi_modopen.9f
1508 ddi_modsym.9f               := LINKSRC = ddi_modopen.9f

1510 ddi_peek16.9f                := LINKSRC = ddi_peek.9f
1511 ddi_peek32.9f                := LINKSRC = ddi_peek.9f
1512 ddi_peek64.9f              := LINKSRC = ddi_peek.9f
1513 ddi_peek8.9f                := LINKSRC = ddi_peek.9f
1514 ddi_peekc.9f                := LINKSRC = ddi_peek.9f
1515 ddi_peekd.9f                := LINKSRC = ddi_peek.9f
1516 ddi_peekl.9f                := LINKSRC = ddi_peek.9f
1517 ddi_peeks.9f                := LINKSRC = ddi_peek.9f

1519 ddi_poke16.9f                := LINKSRC = ddi_poke.9f
1520 ddi_poke32.9f                := LINKSRC = ddi_poke.9f
1521 ddi_poke64.9f                := LINKSRC = ddi_poke.9f
1522 ddi_poke8.9f                := LINKSRC = ddi_poke.9f
1523 ddi_pokec.9f                := LINKSRC = ddi_poke.9f
1524 ddi_poked.9f                := LINKSRC = ddi_poke.9f
1525 ddi_pokel.9f                := LINKSRC = ddi_poke.9f
1526 ddi_pokes.9f                := LINKSRC = ddi_poke.9f

1528 ddi_prop_modify.9f           := LINKSRC = ddi_prop_create.9f
1529 ddi_prop_remove.9f           := LINKSRC = ddi_prop_create.9f
1530 ddi_prop_remove_all.9f       := LINKSRC = ddi_prop_create.9f
1531 ddi_prop_undefine.9f         := LINKSRC = ddi_prop_create.9f

1533 ddi_prop_get_int64.9f       := LINKSRC = ddi_prop_get_int.9f

1535 ddi_prop_free.9f             := LINKSRC = ddi_prop_lookup.9f
1536 ddi_prop_lookup_byte_array.9f := LINKSRC = ddi_prop_lookup.9f
1537 ddi_prop_lookup_int64_array.9f := LINKSRC = ddi_prop_lookup.9f
1538 ddi_prop_lookup_int_array.9f := LINKSRC = ddi_prop_lookup.9f
1539 ddi_prop_lookup_string.9f     := LINKSRC = ddi_prop_lookup.9f
1540 ddi_prop_lookup_string_array.9f := LINKSRC = ddi_prop_lookup.9f

1542 ddi_getlongprop.9f           := LINKSRC = ddi_prop_op.9f
1543 ddi_getlongprop_buf.9f       := LINKSRC = ddi_prop_op.9f
1544 ddi_getprop.9f               := LINKSRC = ddi_prop_op.9f
1545 ddi_getpropflen.9f           := LINKSRC = ddi_prop_op.9f

1547 ddi_prop_update_byte_array.9f := LINKSRC = ddi_prop_update.9f
1548 ddi_prop_update_int.9f       := LINKSRC = ddi_prop_update.9f
1549 ddi_prop_update_int64.9f     := LINKSRC = ddi_prop_update.9f
1550 ddi_prop_update_int64_array.9f := LINKSRC = ddi_prop_update.9f
1551 ddi_prop_update_int_array.9f := LINKSRC = ddi_prop_update.9f
1552 ddi_prop_update_string.9f    := LINKSRC = ddi_prop_update.9f
1553 ddi_prop_update_string_array.9f := LINKSRC = ddi_prop_update.9f

1555 ddi_put16.9f                 := LINKSRC = ddi_put8.9f
1556 ddi_put32.9f                 := LINKSRC = ddi_put8.9f
1557 ddi_put64.9f                 := LINKSRC = ddi_put8.9f
1558 ddi_putb.9f                  := LINKSRC = ddi_put8.9f
1559 ddi_putl.9f                  := LINKSRC = ddi_put8.9f
1560 ddi_putll.9f                 := LINKSRC = ddi_put8.9f
1561 ddi_putw.9f                  := LINKSRC = ddi_put8.9f

1563 ddi_rep_get16.9f             := LINKSRC = ddi_rep_get8.9f
1564 ddi_rep_get32.9f             := LINKSRC = ddi_rep_get8.9f
1565 ddi_rep_get64.9f             := LINKSRC = ddi_rep_get8.9f
1566 ddi_rep_getb.9f              := LINKSRC = ddi_rep_get8.9f
1567 ddi_rep_getl.9f              := LINKSRC = ddi_rep_get8.9f
1568 ddi_rep_getll.9f             := LINKSRC = ddi_rep_get8.9f
1569 ddi_rep_getw.9f              := LINKSRC = ddi_rep_get8.9f

```

```

1571 ddi_rep_put16.9f      := LINKSRC = ddi_rep_put8.9f
1572 ddi_rep_put32.9f     := LINKSRC = ddi_rep_put8.9f
1573 ddi_rep_put64.9f     := LINKSRC = ddi_rep_put8.9f
1574 ddi_rep_putb.9f      := LINKSRC = ddi_rep_put8.9f
1575 ddi_rep_putl.9f      := LINKSRC = ddi_rep_put8.9f
1576 ddi_rep_putll.9f     := LINKSRC = ddi_rep_put8.9f
1577 ddi_rep_putw.9f      := LINKSRC = ddi_rep_put8.9f

1579 ddi_segmap_setup.9f  := LINKSRC = ddi_segmap.9f

1581 ddi_get_soft_state.9f := LINKSRC = ddi_soft_state.9f
1582 ddi_soft_state_fini.9f := LINKSRC = ddi_soft_state.9f
1583 ddi_soft_state_free.9f := LINKSRC = ddi_soft_state.9f
1584 ddi_soft_state_init.9f := LINKSRC = ddi_soft_state.9f
1585 ddi_soft_state_zalloc.9f := LINKSRC = ddi_soft_state.9f

1587 ddi_strtoull.9f      := LINKSRC = ddi_strtoll.9f

1589 ddi_umem_free.9f     := LINKSRC = ddi_umem_alloc.9f

1591 ddi_umem_unlock.9f   := LINKSRC = ddi_umem_lock.9f

1593 devmap_umem_setup.9f := LINKSRC = devmap_devmem_setup.9f

1595 ddi_devmap_segmap.9f := LINKSRC = devmap_setup.9f

1597 devmap_load.9f      := LINKSRC = devmap_unload.9f

1599 dlerrorack.9f       := LINKSRC = dlbindack.9f
1600 dlokack.9f          := LINKSRC = dlbindack.9f
1601 dlphysaddrack.9f    := LINKSRC = dlbindack.9f
1602 dluderrorind.9f     := LINKSRC = dlbindack.9f

1604 desballoc.9f        := LINKSRC = esballoc.9f

1606 unfreezestr.9f      := LINKSRC = freezestr.9f

1608 free_pktiopb.9f     := LINKSRC = get_pktiopb.9f

1610 gld_intr.9f         := LINKSRC = gld.9f
1611 gld_mac_alloc.9f    := LINKSRC = gld.9f
1612 gld_mac_free.9f    := LINKSRC = gld.9f
1613 gld_recv.9f        := LINKSRC = gld.9f
1614 gld_register.9f    := LINKSRC = gld.9f
1615 gld_sched.9f       := LINKSRC = gld.9f
1616 gld_unregister.9f  := LINKSRC = gld.9f

1618 id32_free.9f        := LINKSRC = id32_alloc.9f
1619 id32_lookup.9f     := LINKSRC = id32_alloc.9f

1621 inl.9f              := LINKSRC = inb.9f
1622 inw.9f              := LINKSRC = inb.9f
1623 repinsb.9f          := LINKSRC = inb.9f
1624 repinsd.9f          := LINKSRC = inb.9f
1625 repinsw.9f          := LINKSRC = inb.9f

1627 kmem_free.9f       := LINKSRC = kmem_alloc.9f
1628 kmem_zalloc.9f     := LINKSRC = kmem_alloc.9f

1630 kmem_cache_alloc.9f := LINKSRC = kmem_cache_create.9f
1631 kmem_cache_destroy.9f := LINKSRC = kmem_cache_create.9f
1632 kmem_cache_free.9f := LINKSRC = kmem_cache_create.9f
1633 kmem_cache_set_move.9f := LINKSRC = kmem_cache_create.9f

1635 kstat_named_setstr.9f := LINKSRC = kstat_named_init.9f

```

```

1637 kstat_runq_back_to_waitq.9f := LINKSRC = kstat_queue.9f
1638 kstat_runq_enter.9f         := LINKSRC = kstat_queue.9f
1639 kstat_runq_exit.9f          := LINKSRC = kstat_queue.9f
1640 kstat_waitq_enter.9f        := LINKSRC = kstat_queue.9f
1641 kstat_waitq_exit.9f         := LINKSRC = kstat_queue.9f
1642 kstat_waitq_to_runq.9f     := LINKSRC = kstat_queue.9f

1644 ldi_awrite.9f              := LINKSRC = ldi_aread.9f

1646 ldi_get_devid.9f           := LINKSRC = ldi_get_dev.9f
1647 ldi_get_minor_name.9f      := LINKSRC = ldi_get_dev.9f
1648 ldi_get_otyp.9f            := LINKSRC = ldi_get_dev.9f

1650 ldi_ident_from_dip.9f       := LINKSRC = ldi_ident_from_dev.9f
1651 ldi_ident_from_stream.9f    := LINKSRC = ldi_ident_from_dev.9f
1652 ldi_ident_release.9f       := LINKSRC = ldi_ident_from_dev.9f

1654 ldi_close.9f               := LINKSRC = ldi_open_by_dev.9f
1655 ldi_open_by_devid.9f        := LINKSRC = ldi_open_by_dev.9f
1656 ldi_open_by_name.9f        := LINKSRC = ldi_open_by_dev.9f

1658 ldi_prop_get_int64.9f      := LINKSRC = ldi_prop_get_int.9f

1660 ldi_prop_lookup_byte_array.9f := LINKSRC = ldi_prop_lookup_int_array.9f
1661 ldi_prop_lookup_int64_array.9f := LINKSRC = ldi_prop_lookup_int_array.9f
1662 ldi_prop_lookup_string.9f    := LINKSRC = ldi_prop_lookup_int_array.9f
1663 ldi_prop_lookup_string_array.9f := LINKSRC = ldi_prop_lookup_int_array.9f

1665 ldi_getmsg.9f              := LINKSRC = ldi_putmsg.9f

1667 ldi_write.9f               := LINKSRC = ldi_read.9f

1669 list_destroy.9f            := LINKSRC = list_create.9f
1670 list_head.9f               := LINKSRC = list_create.9f
1671 list_insert_after.9f        := LINKSRC = list_create.9f
1672 list_insert_before.9f       := LINKSRC = list_create.9f
1673 list_insert_head.9f         := LINKSRC = list_create.9f
1674 list_insert_tail.9f         := LINKSRC = list_create.9f
1675 list_is_empty.9f           := LINKSRC = list_create.9f
1676 list_link_active.9f         := LINKSRC = list_create.9f
1677 list_link_init.9f           := LINKSRC = list_create.9f
1678 list_link_replace.9f        := LINKSRC = list_create.9f
1679 list_move_tail.9f           := LINKSRC = list_create.9f
1680 list_next.9f                := LINKSRC = list_create.9f
1681 list_prev.9f                := LINKSRC = list_create.9f
1682 list_remove.9f              := LINKSRC = list_create.9f
1683 list_remove_head.9f         := LINKSRC = list_create.9f
1684 list_remove_tail.9f        := LINKSRC = list_create.9f
1685 list_tail.9f                := LINKSRC = list_create.9f

1687 makecom_g0.9f              := LINKSRC = makecom.9f
1688 makecom_g0_s.9f            := LINKSRC = makecom.9f
1689 makecom_g1.9f              := LINKSRC = makecom.9f
1690 makecom_g5.9f              := LINKSRC = makecom.9f

1692 membar_consumer.9f         := LINKSRC = membar_ops.9f
1693 membar_enter.9f           := LINKSRC = membar_ops.9f
1694 membar_exit.9f             := LINKSRC = membar_ops.9f
1695 membar_producer.9f         := LINKSRC = membar_ops.9f

1697 memcmp.9f                  := LINKSRC = memchr.9f
1698 memcpy.9f                  := LINKSRC = memchr.9f
1699 memmove.9f                  := LINKSRC = memchr.9f
1700 memset.9f                   := LINKSRC = memchr.9f

```

```

1702 mod_info.9f           := LINKSRC = mod_install.9f
1703 mod_modname.9f        := LINKSRC = mod_install.9f
1704 mod_remove.9f         := LINKSRC = mod_install.9f

1706 mutex_destroy.9f     := LINKSRC = mutex.9f
1707 mutex_enter.9f       := LINKSRC = mutex.9f
1708 mutex_exit.9f        := LINKSRC = mutex.9f
1709 mutex_init.9f        := LINKSRC = mutex.9f
1710 mutex_owmed.9f       := LINKSRC = mutex.9f
1711 mutex_tryenter.9f    := LINKSRC = mutex.9f

1713 net_event_notify_unregister.9f := LINKSRC = net_event_notify_register.9f

1715 net_instance_notify_unregister.9f := LINKSRC = net_instance_notify_registe

1717 net_instance_protocol_unregister.9f := LINKSRC = net_protocol_notify_registe

1719 nvlist_add_boolean_array.9f := LINKSRC = nvlist_add_boolean.9f
1720 nvlist_add_boolean_value.9f := LINKSRC = nvlist_add_boolean.9f
1721 nvlist_add_byte.9f        := LINKSRC = nvlist_add_boolean.9f
1722 nvlist_add_byte_array.9f := LINKSRC = nvlist_add_boolean.9f
1723 nvlist_add_int16.9f       := LINKSRC = nvlist_add_boolean.9f
1724 nvlist_add_int16_array.9f := LINKSRC = nvlist_add_boolean.9f
1725 nvlist_add_int32.9f      := LINKSRC = nvlist_add_boolean.9f
1726 nvlist_add_int32_array.9f := LINKSRC = nvlist_add_boolean.9f
1727 nvlist_add_int64.9f      := LINKSRC = nvlist_add_boolean.9f
1728 nvlist_add_int64_array.9f := LINKSRC = nvlist_add_boolean.9f
1729 nvlist_add_int8.9f       := LINKSRC = nvlist_add_boolean.9f
1730 nvlist_add_int8_array.9f := LINKSRC = nvlist_add_boolean.9f
1731 nvlist_add_nvlist.9f     := LINKSRC = nvlist_add_boolean.9f
1732 nvlist_add_nvlist_array.9f := LINKSRC = nvlist_add_boolean.9f
1733 nvlist_add_nvpair.9f     := LINKSRC = nvlist_add_boolean.9f
1734 nvlist_add_string.9f    := LINKSRC = nvlist_add_boolean.9f
1735 nvlist_add_string_array.9f := LINKSRC = nvlist_add_boolean.9f
1736 nvlist_add_uint16.9f    := LINKSRC = nvlist_add_boolean.9f
1737 nvlist_add_uint16_array.9f := LINKSRC = nvlist_add_boolean.9f
1738 nvlist_add_uint32.9f     := LINKSRC = nvlist_add_boolean.9f
1739 nvlist_add_uint32_array.9f := LINKSRC = nvlist_add_boolean.9f
1740 nvlist_add_uint64.9f     := LINKSRC = nvlist_add_boolean.9f
1741 nvlist_add_uint64_array.9f := LINKSRC = nvlist_add_boolean.9f
1742 nvlist_add_uint8.9f     := LINKSRC = nvlist_add_boolean.9f
1743 nvlist_add_uint8_array.9f := LINKSRC = nvlist_add_boolean.9f
1744 nvlist_t.9f            := LINKSRC = nvlist_add_boolean.9f

1746 nv_alloc_fini.9f       := LINKSRC = nvlist_alloc.9f
1747 nv_alloc_init.9f      := LINKSRC = nvlist_alloc.9f
1748 nvlist_dup.9f         := LINKSRC = nvlist_alloc.9f
1749 nvlist_free.9f        := LINKSRC = nvlist_alloc.9f
1750 nvlist_merge.9f       := LINKSRC = nvlist_alloc.9f
1751 nvlist_pack.9f        := LINKSRC = nvlist_alloc.9f
1752 nvlist_size.9f        := LINKSRC = nvlist_alloc.9f
1753 nvlist_unpack.9f      := LINKSRC = nvlist_alloc.9f
1754 nvlist_xalloc.9f      := LINKSRC = nvlist_alloc.9f
1755 nvlist_xdup.9f        := LINKSRC = nvlist_alloc.9f
1756 nvlist_xpack.9f       := LINKSRC = nvlist_alloc.9f
1757 nvlist_xunpack.9f     := LINKSRC = nvlist_alloc.9f

1759 nvlist_lookup_boolean_array.9f := LINKSRC = nvlist_lookup_boolean.9f
1760 nvlist_lookup_boolean_value.9f := LINKSRC = nvlist_lookup_boolean.9f
1761 nvlist_lookup_byte.9f        := LINKSRC = nvlist_lookup_boolean.9f
1762 nvlist_lookup_byte_array.9f := LINKSRC = nvlist_lookup_boolean.9f
1763 nvlist_lookup_int16.9f       := LINKSRC = nvlist_lookup_boolean.9f
1764 nvlist_lookup_int16_array.9f := LINKSRC = nvlist_lookup_boolean.9f
1765 nvlist_lookup_int32.9f      := LINKSRC = nvlist_lookup_boolean.9f
1766 nvlist_lookup_int32_array.9f := LINKSRC = nvlist_lookup_boolean.9f
1767 nvlist_lookup_int64.9f      := LINKSRC = nvlist_lookup_boolean.9f

```

```

1768 nvlist_lookup_int64_array.9f := LINKSRC = nvlist_lookup_boolean.9f
1769 nvlist_lookup_int8.9f        := LINKSRC = nvlist_lookup_boolean.9f
1770 nvlist_lookup_int8_array.9f := LINKSRC = nvlist_lookup_boolean.9f
1771 nvlist_lookup_nvlist.9f     := LINKSRC = nvlist_lookup_boolean.9f
1772 nvlist_lookup_nvlist_array.9f := LINKSRC = nvlist_lookup_boolean.9f
1773 nvlist_lookup_pairs.9f      := LINKSRC = nvlist_lookup_boolean.9f
1774 nvlist_lookup_string.9f     := LINKSRC = nvlist_lookup_boolean.9f
1775 nvlist_lookup_string_array.9f := LINKSRC = nvlist_lookup_boolean.9f
1776 nvlist_lookup_uint16.9f     := LINKSRC = nvlist_lookup_boolean.9f
1777 nvlist_lookup_uint16_array.9f := LINKSRC = nvlist_lookup_boolean.9f
1778 nvlist_lookup_uint32.9f     := LINKSRC = nvlist_lookup_boolean.9f
1779 nvlist_lookup_uint32_array.9f := LINKSRC = nvlist_lookup_boolean.9f
1780 nvlist_lookup_uint64.9f     := LINKSRC = nvlist_lookup_boolean.9f
1781 nvlist_lookup_uint64_array.9f := LINKSRC = nvlist_lookup_boolean.9f
1782 nvlist_lookup_uint8.9f      := LINKSRC = nvlist_lookup_boolean.9f
1783 nvlist_lookup_uint8_array.9f := LINKSRC = nvlist_lookup_boolean.9f

1785 nvlist_exists.9f          := LINKSRC = nvlist_lookup_nvpair.9f

1787 nvpair_name.9f           := LINKSRC = nvlist_next_nvpair.9f
1788 nvpair_type.9f          := LINKSRC = nvlist_next_nvpair.9f

1790 nvlist_remove_all.9f     := LINKSRC = nvlist_remove.9f

1792 nvpair_value_boolean_array.9f := LINKSRC = nvpair_value_byte.9f
1793 nvpair_value_byte_array.9f   := LINKSRC = nvpair_value_byte.9f
1794 nvpair_value_int16.9f        := LINKSRC = nvpair_value_byte.9f
1795 nvpair_value_int16_array.9f := LINKSRC = nvpair_value_byte.9f
1796 nvpair_value_int32.9f       := LINKSRC = nvpair_value_byte.9f
1797 nvpair_value_int32_array.9f := LINKSRC = nvpair_value_byte.9f
1798 nvpair_value_int64.9f       := LINKSRC = nvpair_value_byte.9f
1799 nvpair_value_int64_array.9f := LINKSRC = nvpair_value_byte.9f
1800 nvpair_value_int8.9f        := LINKSRC = nvpair_value_byte.9f
1801 nvpair_value_int8_array.9f   := LINKSRC = nvpair_value_byte.9f
1802 nvpair_value_nvlist.9f      := LINKSRC = nvpair_value_byte.9f
1803 nvpair_value_nvlist_array.9f := LINKSRC = nvpair_value_byte.9f
1804 nvpair_value_string.9f       := LINKSRC = nvpair_value_byte.9f
1805 nvpair_value_string_array.9f := LINKSRC = nvpair_value_byte.9f
1806 nvpair_value_uint16.9f      := LINKSRC = nvpair_value_byte.9f
1807 nvpair_value_uint16_array.9f := LINKSRC = nvpair_value_byte.9f
1808 nvpair_value_uint32.9f      := LINKSRC = nvpair_value_byte.9f
1809 nvpair_value_uint32_array.9f := LINKSRC = nvpair_value_byte.9f
1810 nvpair_value_uint64.9f      := LINKSRC = nvpair_value_byte.9f
1811 nvpair_value_uint64_array.9f := LINKSRC = nvpair_value_byte.9f
1812 nvpair_value_uint8.9f       := LINKSRC = nvpair_value_byte.9f
1813 nvpair_value_uint8_array.9f := LINKSRC = nvpair_value_byte.9f

1815 outl.9f                  := LINKSRC = outb.9f
1816 outw.9f                  := LINKSRC = outb.9f
1817 repoutsb.9f              := LINKSRC = outb.9f
1818 repoutsd.9f              := LINKSRC = outb.9f
1819 repoutsw.9f              := LINKSRC = outb.9f

1821 pci_config_get16.9f       := LINKSRC = pci_config_get8.9f
1822 pci_config_get32.9f       := LINKSRC = pci_config_get8.9f
1823 pci_config_get64.9f       := LINKSRC = pci_config_get8.9f
1824 pci_config_getb.9f       := LINKSRC = pci_config_get8.9f
1825 pci_config_getl.9f       := LINKSRC = pci_config_get8.9f
1826 pci_config_getll.9f      := LINKSRC = pci_config_get8.9f
1827 pci_config_getw.9f       := LINKSRC = pci_config_get8.9f
1828 pci_config_put16.9f      := LINKSRC = pci_config_get8.9f
1829 pci_config_put32.9f      := LINKSRC = pci_config_get8.9f
1830 pci_config_put64.9f      := LINKSRC = pci_config_get8.9f
1831 pci_config_put8.9f        := LINKSRC = pci_config_get8.9f
1832 pci_config_putb.9f       := LINKSRC = pci_config_get8.9f
1833 pci_config_putl.9f       := LINKSRC = pci_config_get8.9f

```

```

1834 pci_config_putll.9f      := LINKSRC = pci_config_get8.9f
1835 pci_config_putw.9f      := LINKSRC = pci_config_get8.9f

1837 pci_config_takedown.9f  := LINKSRC = pci_config_setup.9f

1839 pci_ereport_post.9f     := LINKSRC = pci_ereport_setup.9f
1840 pci_ereport_takedown.9f := LINKSRC = pci_ereport_setup.9f

1842 pci_restore_config_regs.9f := LINKSRC = pci_save_config_regs.9f

1844 minphys.9f             := LINKSRC = physio.9f

1846 pm_idle_component.9f    := LINKSRC = pm_busy_component.9f

1848 pm_lower_power.9f      := LINKSRC = pm_raise_power.9f

1850 priv_policy_choice.9f   := LINKSRC = priv_policy.9f
1851 priv_policy_only.9f    := LINKSRC = priv_policy.9f

1853 proc_ref.9f            := LINKSRC = proc_signal.9f
1854 proc_unref.9f         := LINKSRC = proc_signal.9f

1856 qprocsoff.9f          := LINKSRC = qprocson.9f

1858 qwait_sig.9f           := LINKSRC = qwait.9f

1860 rmallocmap_wait.9f     := LINKSRC = rmallocmap.9f
1861 rmfreemap.9f          := LINKSRC = rmallocmap.9f

1863 rw_destroy.9f          := LINKSRC = rwlock.9f
1864 rw_downgrade.9f       := LINKSRC = rwlock.9f
1865 rw_enter.9f           := LINKSRC = rwlock.9f
1866 rw_exit.9f            := LINKSRC = rwlock.9f
1867 rw_init.9f            := LINKSRC = rwlock.9f
1868 rw_read_locked.9f     := LINKSRC = rwlock.9f
1869 rw_tryenter.9f        := LINKSRC = rwlock.9f
1870 rw_tryupgrade.9f      := LINKSRC = rwlock.9f

1872 scsi_dname.9f          := LINKSRC = scsi_cname.9f
1873 scsi_mname.9f         := LINKSRC = scsi_cname.9f
1874 scsi_rname.9f         := LINKSRC = scsi_cname.9f
1875 scsi_sname.9f         := LINKSRC = scsi_cname.9f

1877 scsi_dmafree.9f       := LINKSRC = scsi_dmaget.9f

1879 scsi_sense_cmdspecific_uint64.9f := LINKSRC = scsi_ext_sense_fields.9f
1880 scsi_sense_info_uint64.9f := LINKSRC = scsi_ext_sense_fields.9f

1886 scsi_hba_attach.9f    := LINKSRC = scsi_hba_attach_setup.9f
1882 scsi_hba_detach.9f   := LINKSRC = scsi_hba_attach_setup.9f

1884 scsi_hba_fini.9f      := LINKSRC = scsi_hba_init.9f

1886 scsi_hba_pkt_free.9f  := LINKSRC = scsi_hba_pkt_alloc.9f

1888 scsi_hba_tran_free.9f := LINKSRC = scsi_hba_tran_alloc.9f

1890 scsi_ifsetcap.9f     := LINKSRC = scsi_ifgetcap.9f

1892 scsi_pktfree.9f       := LINKSRC = scsi_pktalloc.9f
1893 scsi_resalloc.9f     := LINKSRC = scsi_pktalloc.9f
1894 scsi_resfree.9f      := LINKSRC = scsi_pktalloc.9f

1896 scsi_sense_asc.9f     := LINKSRC = scsi_sense_key.9f
1897 scsi_sense_ascq.9f   := LINKSRC = scsi_sense_key.9f

```

```

1899 scsi_unslave.9f       := LINKSRC = scsi_unprobe.9f

1901 sema_destroy.9f       := LINKSRC = semaphore.9f
1902 sema_init.9f         := LINKSRC = semaphore.9f
1903 sema_p.9f            := LINKSRC = semaphore.9f
1904 sema_p_sig.9f        := LINKSRC = semaphore.9f
1905 sema_tryop.9f        := LINKSRC = semaphore.9f
1906 sema_v.9f            := LINKSRC = semaphore.9f

1908 numtos.9f             := LINKSRC = stoi.9f

1910 ddi_strdup.9f         := LINKSRC = string.9f
1911 strcasecmp.9f        := LINKSRC = string.9f
1912 strchr.9f           := LINKSRC = string.9f
1913 strcmp.9f           := LINKSRC = string.9f
1914 strcpy.9f           := LINKSRC = string.9f
1915 strdup.9f           := LINKSRC = string.9f
1916 strfree.9f          := LINKSRC = string.9f
1917 strlcat.9f          := LINKSRC = string.9f
1918 strlcpy.9f          := LINKSRC = string.9f
1919 strlen.9f           := LINKSRC = string.9f
1920 strncasecmp.9f      := LINKSRC = string.9f
1921 strncat.9f          := LINKSRC = string.9f
1922 strncmp.9f          := LINKSRC = string.9f
1923 strncpy.9f          := LINKSRC = string.9f
1924 strnlen.9f          := LINKSRC = string.9f
1925 strrchr.9f          := LINKSRC = string.9f
1926 strspn.9f           := LINKSRC = string.9f

1928 ddi_taskq_create.9f   := LINKSRC = taskq.9f
1929 ddi_taskq_destroy.9f := LINKSRC = taskq.9f
1930 ddi_taskq_dispatch.9f := LINKSRC = taskq.9f
1931 ddi_taskq_resume.9f  := LINKSRC = taskq.9f
1932 ddi_taskq_suspend.9f := LINKSRC = taskq.9f
1933 ddi_taskq_wait.9f   := LINKSRC = taskq.9f
1934 taskq_suspended.9f  := LINKSRC = taskq.9f

1936 uconv_u16tou8.9f     := LINKSRC = uconv_u16tou32.9f
1937 uconv_u32tou16.9f   := LINKSRC = uconv_u16tou32.9f
1938 uconv_u32tou8.9f    := LINKSRC = uconv_u16tou32.9f
1939 uconv_u8tou16.9f    := LINKSRC = uconv_u16tou32.9f
1940 uconv_u8tou32.9f    := LINKSRC = uconv_u16tou32.9f

1942 usb_alloc_bulk_req.9f := LINKSRC = usb_alloc_request.9f
1943 usb_alloc_ctrl_req.9f := LINKSRC = usb_alloc_request.9f
1944 usb_alloc_intr_req.9f := LINKSRC = usb_alloc_request.9f
1945 usb_alloc_isoc_req.9f := LINKSRC = usb_alloc_request.9f
1946 usb_free_bulk_req.9f := LINKSRC = usb_alloc_request.9f
1947 usb_free_ctrl_req.9f := LINKSRC = usb_alloc_request.9f
1948 usb_free_intr_req.9f := LINKSRC = usb_alloc_request.9f
1949 usb_free_isoc_req.9f := LINKSRC = usb_alloc_request.9f
1950 usb_client_detach.9f := LINKSRC = usb_client_attach.9f

1952 usb_get_if_number.9f  := LINKSRC = usb_get_alt_if.9f
1953 usb_owns_device.9f  := LINKSRC = usb_get_alt_if.9f
1954 usb_set_alt_if.9f   := LINKSRC = usb_get_alt_if.9f

1956 usb_set_cfg.9f       := LINKSRC = usb_get_cfg.9f

1958 usb_free_descr_tree.9f := LINKSRC = usb_get_dev_data.9f
1959 usb_free_dev_data.9f  := LINKSRC = usb_get_dev_data.9f
1960 usb_print_descr_tree.9f := LINKSRC = usb_get_dev_data.9f

1962 usb_pipe_ctrl_xfer_wait.9f := LINKSRC = usb_pipe_ctrl_xfer.9f
1963 usb_pipe_stop_intr_polling.9f := LINKSRC = usb_pipe_intr_xfer.9f
1964 usb_pipe_stop_isoc_polling.9f := LINKSRC = usb_pipe_isoc_xfer.9f

```

```
1966 usb_pipe_get_private.9f      := LINKSRC = usb_pipe_set_private.9f
1968 usb_unregister_hotplug_cbs.9f := LINKSRC = usb_register_hotplug_cbs.9f
1970 va_copy.9f                   := LINKSRC = va_arg.9f
1971 va_end.9f                    := LINKSRC = va_arg.9f
1972 va_start.9f                  := LINKSRC = va_arg.9f
1974 .KEEP_STATE:
1976 include                       $(SRC)/man/Makefile.man
1978 install:                      $(ROOTMANFILES) $(ROOTMANLINKS)
```

```

*****
5192 Sat May 24 17:48:26 2014
new/usr/src/man/man9f/ddi_check_acc_handle.9f
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dma_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 \" te
2.\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
3.\" Copyright (c) 1999, Sun Microsystems, Inc. All Rights Reserved
4.\" The contents of this file are subject to the terms of the Common Development
5.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
6.\" When distributing Covered Code, include this CDDL HEADER in each file and in
7.TH DDI_CHECK_ACC_HANDLE 9F "May 24, 2014"
8.TH DDI_CHECK_ACC_HANDLE 9F "Aug 13, 1999"
9.SH NAME
10 ddi_check_acc_handle, ddi_check_dma_handle \- Check data access and DMA handles
11.SH SYNOPSIS
12.LP
13.nf
14 #include <sys/ddi.h>
15 #include <sys/sunddi.h>

18 \fBint\fR \fBddi_check_acc_handle\fR(\fBddi_acc_handle_t\fR \fI acc_handle\fR );
19 .fi

21.LP
22.nf
23 \fBint\fR \fBddi_check_dma_handle\fR(\fBddi_dma_handle_t\fR \fI dma_handle\fR );
24 .fi

26.SH INTERFACE LEVEL
27.sp
28.LP
29 Solaris DDI specific (Solaris DDI)
30.SH PARAMETERS
31.sp
32.ne 2
33.na
34 \fB\fIacc_handle\fR \fR
35.ad
36.RS 15n
37 Data access handle obtained from a previous call to
38 \fBddi_regs_map_setup\fR(9F), \fBddi_dma_mem_alloc\fR(9F), or similar function.
39.RE

41.sp
42.ne 2
43.na
44 \fB\fIdma_handle\fR \fR
45.ad
46.RS 15n
47 DMA handle obtained from \fBddi_dma_alloc_handle\fR(9F).
48 DMA handle obtained from a previous call to \fBddi_dma_setup\fR(9F) or one of
49 its derivatives.
50.RE

50.SH DESCRIPTION
51.sp
52.LP
53 The \fBddi_check_acc_handle()\fR and \fBddi_check_dma_handle()\fR functions

```

```

54 check for faults that can interfere with communication between a driver and the
55 device it controls. Each function checks a single handle of a specific type and
56 returns a status value indicating whether faults affecting the resource mapped
57 by the supplied handle have been detected.
58.sp
59.LP
60 If a fault is indicated when checking a data access handle, this implies that
61 the driver is no longer able to access the mapped registers or memory using
62 programmed I/O through that handle. Typically, this might occur after the
63 device has failed to respond to an I/O access (for example, has incurred a bus
64 error or timed out). The effect of programmed I/O accesses made after this
65 happens is undefined; for example, read accesses (for example,
66 \fBddi_get8\fR(9F)) may return random values, and write accesses (for example,
67 \fBddi_put8\fR(9F)) may or may not have any effect. This type of fault is
68 normally fatal to the operation of the device, and the driver should report it
69 via \fBddi_dev_report_fault\fR(9F) specifying \fBDDI_SERVICE_LOST\fR for the
70 impact, and \fBDDI_DATAPATH_FAULT\fR for the location.
71.sp
72.LP
73 If a fault is indicated when checking a DMA handle, it implies that a fault has
74 been detected that has (or will) affect DMA transactions between the device and
75 the memory currently bound to the handle (or most recently bound, if the handle
76 is currently unbound). Possible causes include the failure of a component in
77 the DMA data path, or an attempt by the device to make an invalid DMA access.
78 The driver may be able to continue by falling back to a non-DMA mode of
79 operation, but in general, DMA faults are non-recoverable. The contents of the
80 memory currently (or previously) bound to the handle should be regarded as
81 indeterminate. The fault indication associated with the current transaction is
82 lost once the handle is (re-)bound, but because the fault may persist, future
83 DMA operations may not succeed.
84.sp
85.LP
86 Note that some implementations cannot detect all types of failure. If a fault
87 is not indicated, this does not constitute a guarantee that communication is
88 possible. However, if a check fails, this is a positive indication that a
89 problem \fBdoes\fR exist with respect to communication using that handle.
90.SH RETURN VALUES
91.sp
92.LP
93 The \fBddi_check_acc_handle()\fR and \fBddi_check_dma_handle()\fR functions
94 return \fBDDI_SUCCESS\fR if no faults affecting the supplied handle are
95 detected and \fBDDI_FAILURE\fR if any fault affecting the supplied handle is
96 detected.
97.SH EXAMPLES
98.sp
99.in +2
100.nf
101 static int
102 xxattach(dev_info_t *dip, ddi_attach_cmd_t cmd)
103 {
104     \&...
105     /* This driver uses only a single register-access handle */
106     status = ddi_regs_map_setup(dip, REGSET_ZERO, &regaddr,
107                                0, 0, , &acc_attrs, &acc_hdl);
108     if (status != DDI_SUCCESS)
109         return (DDI_FAILURE);
110     \&...
111 }

113 static int
114 xxread(dev_t dev, struct uio *uio_p, cred_t *cred_p)
115 {
116     \&...
117     if (ddi_check_acc_handle(acc_hdl) != DDI_SUCCESS) {
118         ddi_dev_report_fault(dip, DDI_SERVICE_LOST,
119                               DDI_DATAPATH_FAULT, "register access fault during read");

```



```
120         return (EIO);
121     }
122     \&...
123 .fi
124 .in -2

126 .SH CONTEXT
127 .sp
128 .LP
129 The \fBddi_check_acc_handle()\fR and \fBddi_check_dma_handle()\fR functions may
130 be called from user, kernel, or interrupt context.
131 .SH SEE ALSO
132 .sp
133 .LP
134 \fBddi_regs_map_setup\fR(9F), \fBddi_dma_alloc_handle\fR(9F),
134 \fBddi_regs_map_setup\fR(9F), \fBddi_dma_setup\fR(9F),
135 \fBddi_dev_report_fault\fR(9F), \fBddi_get8\fR(9F), \fBddi_put8\fR(9F)
```

```

*****
2037 Sat May 24 17:48:26 2014
new/usr/src/man/man9f/ddi_dma_burstsizes.9f
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1  \" te
2  \. Copyright 2014 Garrett D'Amore <garrett@damore.org>
3  \. Copyright (c) 1994, Sun Microsystems, Inc.
4  \. The contents of this file are subject to the terms of the Common Development
5  \. You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
6  \. When distributing Covered Code, include this CDDL HEADER in each file and in
7  .TH DDI_DMA_BURSTSIZES 9F "May 24, 2014"
8  .TH DDI_DMA_BURSTSIZES 9F "Feb 1, 1994"
9  .SH NAME
10 ddi_dma_burstsizes \- find out the allowed burst sizes for a DMA mapping
11 .SH SYNOPSIS
12 .LP
13 #include <sys/conf.h>
14 #include <sys/ddi.h>
15 #include <sys/sunddi.h>

19 \fBint\fR \fBddi_dma_burstsizes\fR(\fBddi_dma_handle_t\fR \fIhandle\fR);
20 .fi

22 .SH INTERFACE LEVEL
23 .sp
24 .LP
25 Solaris DDI specific (Solaris DDI).
26 .SH PARAMETERS
27 .sp
28 .ne 2
29 .na
30 \fB\fIhandle\fR \fR
31 .ad
32 .RS 11n
33 A \fBDMA\fR handle.
34 A \fBDMA\fR handle that was filled in by a successful call to
35 \fBddi_dma_setup\fR(9F).
36 .RE

36 .SH DESCRIPTION
37 .sp
38 .LP
39 \fBddi_dma_burstsizes()\fR returns the allowed burst sizes for a \fBDMA\fR
40 mapping. This value is derived from the \fBdma_attr_burstsizes\fR member of the
41 \fBddi_dma_attr\fR(9S) structure, but it shows the allowable burstsizes
42 mapping. This value is derived from the \fBdlim_burstsizes\fR member of the
43 \fBddi_dma_lim_sparc\fR(9S) structure, but it shows the allowable burstsizes
44 \fBafter\fR imposing on it the limitations of other device layers in addition
45 to device's own limitations.
46 .SH RETURN VALUES
47 .sp
48 .LP
49 \fBddi_dma_burstsizes()\fR returns a binary encoded value of the allowable
50 \fBDMA\fR burst sizes. See \fBddi_dma_attr\fR(9S) for a discussion of
51 \fBDMA\fR burst sizes. See \fBddi_dma_lim_sparc\fR(9S) for a discussion of
52 \fBDMA\fR burst sizes.
53 .SH CONTEXT

```

```

51 .sp
52 .LP
53 This function can be called from user or interrupt context.
54 .SH SEE ALSO
55 .sp
56 .LP
57 \fBddi_dma_alloc_handle\fR(9F), \fBddi_dma_attr\fR(9S)
58 \fBddi_dma_devalign\fR(9F), \fBddi_dma_setup\fR(9F),
59 \fBddi_dma_lim_sparc\fR(9S), \fBddi_dma_req\fR(9S)
60 .sp
61 .LP
62 \fIWriting Device Drivers\fR

```

```

*****
11099 Sat May 24 17:48:26 2014
new/usr/src/man/man9f/ddi_dmae.9f
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 \" te
2 .\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
3 .\" Copyright (c) 2006 Sun Microsystems, Inc. All Rights Reserved.
4 .\" Copyright 2012 Garrett D'Amore <garrett@damore.org>. 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 DDI_DMAE_9F \"May 24, 2014\"
9 .TH DDI_DMAE_9F \"Feb 02, 2012\"
10 .SH NAME
11 ddi_dmae, ddi_dmae_alloc, ddi_dmae_release, ddi_dmae_prog, ddi_dmae_disable,
12 ddi_dmae_enable, ddi_dmae_stop, ddi_dmae_getcnt, ddi_dmae_lstpary,
13 ddi_dmae_getattr \- system DMA engine functions
14 ddi_dmae_getlim, ddi_dmae_getattr \- system DMA engine functions
15 .SH SYNOPSIS
16 .LP
17 .nf
18 \fBint\fR \fBddi_dmae_alloc\fR(\fBdev_info_t * \fR\fIdip\fR, \fBint\fR \fIchnl\fR
19 \fBcaddr_t \fR \fIarg\fR);
20 .fi
21 .LP
22 .nf
23 \fBint\fR \fBddi_dmae_release\fR(\fBdev_info_t * \fR\fIdip\fR, \fBint\fR \fIchnl\
24 .fi
25 .LP
26 .nf
27 \fBint\fR \fBddi_dmae_prog\fR(\fBdev_info_t * \fR\fIdip\fR, \fBstruct ddi_dmae_re
28 \fBddi_dma_cookie_t * \fR\fIcookiep\fR, \fBint\fR \fIchnl\fR);
29 .fi
30 .LP
31 .nf
32 \fBint\fR \fBddi_dmae_disable\fR(\fBdev_info_t * \fR\fIdip\fR, \fBint\fR \fIchnl\
33 .fi
34 .LP
35 .nf
36 \fBint\fR \fBddi_dmae_enable\fR(\fBdev_info_t * \fR\fIdip\fR, \fBint\fR \fIchnl\f
37 .fi
38 .LP
39 .nf
40 \fBint\fR \fBddi_dmae_stop\fR(\fBdev_info_t * \fR\fIdip\fR, \fBint\fR \fIchnl\fR)
41 .fi
42 .LP
43 .nf
44 \fBint\fR \fBddi_dmae_getcnt\fR(\fBdev_info_t * \fR\fIdip\fR, \fBint\fR \fIchnl\
45 .fi
46 .LP
47 .nf
48 \fBint\fR \fBddi_dmae_getattr\fR(\fBdev_info_t * \fR\fIdip\fR, \fBint\fR \fIchnl\
49 .fi
50 .LP
51 .nf
52 \fBint\fR \fBddi_dmae_lstpary\fR(\fBdev_info_t * \fR\fIdip\fR, \fBint\fR \fIchnl
53 .fi

```

```

54 .LP
55 .nf
56 \fBint\fR \fBddi_dmae_getlim\fR(\fBdev_info_t * \fR\fIdip\fR, \fBddi_dma_lim_t * \
57 .fi
58 .LP
59 .nf
60 \fBint\fR \fBddi_dmae_getattr\fR(\fBdev_info_t * \fR\fIdip\fR, \fBddi_dma_attr_t
61 .fi
62 .SH INTERFACE LEVEL
63 .sp
64 .ne 2
65 .na
66 \fB\fIdip\fR\fR
67 .ad
68 .RS 12n
69 A \fBdev_info\fR pointer that identifies the device.
70 .RE
71 .sp
72 .ne 2
73 .na
74 \fB\fIchnl\fR\fR
75 .ad
76 .RS 12n
77 A \fBDMA\fR channel number. On \fBISA\fR buses this number must be \fB0\fR,
78 \fB1\fR, \fB2\fR, \fB3\fR, \fB5\fR, \fB6\fR, or \fB7\fR.
79 .RE
80 .sp
81 .ne 2
82 .na
83 \fB\fIcallback\fR\fR
84 .ad
85 .RS 12n
86 The address of a function to call back later if resources are not currently
87 available. The following special function addresses may also be used:
88 .sp
89 .ne 2
90 .na
91 \fB\fBDDI_DMA_SLEEP\fR\fR
92 .ad
93 .RS 20n
94 Wait until resources are available.
95 .RE
96 .sp
97 .ne 2
98 .na
99 \fB\fBDDI_DMA_DONTWAIT\fR\fR
100 .ad
101 .RS 20n
102 Do not wait until resources are available and do not schedule a callback.
103 .RE
104 .RE

```

```

112 .sp
113 .ne 2
114 .na
115 \fB\fIarg\fR\fR
116 .ad
117 .RS 12n
118 Argument to be passed to the callback function, if specified.
119 .RE

121 .sp
122 .ne 2
123 .na
124 \fB\fIdmaereq\fR\fR
125 .ad
126 .RS 12n
127 A pointer to a \fBDMA\fR engine request structure. See \fBddi_dmae_req\fR(9S).
128 .RE

130 .sp
131 .ne 2
132 .na
133 \fB\fIcookiep\fR\fR
134 .ad
135 .RS 12n
136 A pointer to a \fBddi_dma_cookie\fR(9S) object,
137 which contains the address and count.
138 .RE

140 .sp
141 .ne 2
142 .na
143 \fB\fIcountp\fR\fR
144 .ad
145 .RS 12n
146 A pointer to an integer that will receive the count of the number of bytes not
147 yet transferred upon completion of a \fBDMA\fR operation.
148 .RE

150 .sp
151 .ne 2
152 .na
160 \fB\fIlimitsp\fR\fR
161 .ad
162 .RS 12n
163 A pointer to a \fBDMA\fR limit structure. See \fBddi_dma_lim_x86\fR(9S).
164 .RE

166 .sp
167 .ne 2
168 .na
153 \fB\fIattrp\fR\fR
154 .ad
155 .RS 12n
156 A pointer to a \fBDMA \fR attribute structure. See \fBddi_dma_attr\fR(9S).
157 .RE

159 .SH DESCRIPTION
160 .sp
161 .LP
162 There are three possible ways that a device can perform \fBDMA\fR engine
163 functions:
164 .sp
165 .ne 2
166 .na
167 \fBBus master DMA\fR
168 .ad

```

```

169 .RS 19n
170 If the device is capable of acting as a true bus master, then the driver should
171 program the device's \fBDMA\fR registers directly and not make use of the
172 \fBDMA\fR engine functions described here. The driver should obtain the
173 \fBDMA\fR address and count from \fBddi_dma_cookie\fR(9S).
174 .RE

176 .sp
177 .ne 2
178 .na
179 \fBThird-party \fBDMA\fR\fR
180 .ad
181 .RS 19n
182 This method uses the system \fBDMA\fR engine that is resident on the main
183 system board. In this model, the device cooperates with the system's \fBDMA\fR
184 engine to effect the data transfers between the device and memory. The driver
185 uses the functions documented here, except \fBddi_dmae_lstparty()\fR, to
186 initialize and program the \fBDMA\fR engine. For each \fBDMA\fR data transfer,
187 the driver programs the \fBDMA\fR engine and then gives the device a command
188 to initiate the transfer in cooperation with that engine.
189 .RE

191 .sp
192 .ne 2
193 .na
194 \fBFirst-party DMA\fR
195 .ad
196 .RS 19n
197 Using this method, the device uses its own \fBDMA\fR bus cycles, but requires a
198 channel from the system's \fBDMA\fR engine. After allocating the \fBDMA\fR
199 channel, the \fBddi_dmae_lstparty()\fR function may be used to perform whatever
200 configuration is necessary to enable this mode.
201 .RE

203 .SS "\fBddi_dmae_alloc()\fR"
204 .sp
205 .LP
206 The \fBddi_dmae_alloc()\fR function is used to acquire a \fBDMA\fR channel of
207 the system \fBDMA\fR engine. \fBddi_dmae_alloc()\fR allows only one device at a
208 time to have a particular \fBDMA\fR channel allocated. It must be called prior
209 to any other system \fBDMA\fR engine function on a channel. If the device
210 allows the channel to be shared with other devices, it must be freed using
211 \fBddi_dmae_release()\fR after completion of the \fBDMA\fR operation. In any
212 case, the channel must be released before the driver successfully detaches. See
213 \fBdetach\fR(9E). No other driver may acquire the \fBDMA\fR channel until it is
214 released.
215 .sp
216 .LP
217 If the requested channel is not immediately available, the value of
218 \fIcallback\fR determines what action will be taken. If the value of
219 \fIcallback\fR is \fBDDI_DMA_DONTWAIT\fR, \fBddi_dmae_alloc()\fR will return
220 immediately. The value \fBDDI_DMA_SLEEP\fR will cause the thread to sleep and
221 not return until the channel has been acquired. Any other value is assumed to
222 be a callback function address. In that case, \fBddi_dmae_alloc()\fR returns
223 immediately, and when resources might have become available, the callback
224 function is called (with the argument \fIarg\fR) from interrupt context. When
225 the callback function is called, it should attempt to allocate the \fBDMA\fR
226 channel again. If it succeeds or no longer needs the channel, it must return
227 the value \fBDDI_DMA_CALLBACK_DONE\fR. If it tries to allocate the channel but
228 fails to do so, it must return the value \fBDDI_DMA_CALLBACK_RUNOUT\fR. In this
229 case, the callback function is put back on a list to be called again later.
230 .SS "\fBddi_dmae_prog()\fR"
231 .sp
232 .LP
233 The \fBddi_dmae_prog()\fR function programs the \fBDMA\fR channel for a
234 \fBDMA\fR transfer. The \fBddi_dmae_req\fR structure contains all the

```

235 information necessary to set up the channel, except for the memory address and
 236 count. Once the channel has been programmed, subsequent calls to
 237 `\fBddi_dmae_prog()` may specify a value of `\fINULL` for `\fIdmaereq` if
 238 no changes to the programming are required other than the address and count
 239 values. It disables the channel prior to setup, and enables the channel before
 240 returning. The `\fBDMA` address and count are specified by passing
 241 `\fBddi_dmae_prog()` a `\fBDMA` cookie.
 242 Other `\fBDMA` engine parameters are specified by the `\fBDMA` engine request
 243 structure passed in through `\fIdmaereq`. The fields of that structure are
 244 documented in `\fBddi_dmae_req` for additional information.
 245 .sp
 246 .LP
 247 Before using `\fBddi_dmae_prog()`, you must allocate system `\fBDMA`
 248 resources using `\fBDMA` setup functions such as `\fBddi_dma_mem_alloc`
 249 `\fBddi_dma_addr_bind_handle` can then be used to retrieve a cookie which
 250 contains the address and count. Then this cookie is passed to
 251 `\fBddi_dmae_prog()`.
 252 .SS "`\fBddi_dmae_disable()`"
 253 .sp
 254 .LP
 255 The `\fBddi_dmae_disable()` function disables the `\fBDMA` channel so that it
 256 no longer responds to a device's `\fBDMA` service requests.
 257 .SS "`\fBddi_dmae_enable()`"
 258 .sp
 259 .LP
 260 The `\fBddi_dmae_enable()` function enables the `\fBDMA` channel for
 261 operation. This may be used to re-enable the channel after a call to
 262 `\fBddi_dmae_disable()`. The channel is automatically enabled after successful
 263 programming by `\fBddi_dmae_prog()`.
 264 .SS "`\fBddi_dmae_stop()`"
 265 .sp
 266 .LP
 267 The `\fBddi_dmae_stop()` function disables the channel and terminates any
 268 active operation.
 269 .SS "`\fBddi_dmae_getcnt()`"
 270 .sp
 271 .LP
 272 The `\fBddi_dmae_getcnt()` function examines the count register of the
 273 `\fBDMA` channel and sets `\fI*countp` to the number of bytes remaining to be
 274 transferred. The channel is assumed to be stopped.
 275 .SS "`\fBddi_dmae_1stparty()`"
 276 .sp
 277 .LP
 278 In the case of `\fBISA` buses, `\fBddi_dmae_1stparty()` configures a channel
 279 in the system's `\fBDMA` engine to operate in a "slave" ("cascade") mode.
 280 .sp
 281 .LP
 282 When operating in `\fBddi_dmae_1stparty()` mode, the `\fBDMA` channel must
 283 first be allocated using `\fBddi_dmae_alloc()` and then configured using
 284 `\fBddi_dmae_1stparty()`. The driver then programs the device to perform the
 285 I/O, including the necessary `\fBDMA` address and count values obtained from
 286 the `\fBddi_dma_cookie` for additional information.
 287 .SS "`\fBddi_dmae_getlim()`"
 288 .sp
 289 .LP
 290 This function is obsolete. Use `\fBddi_dmae_getattr()`, described below,
 291 instead.
 292 .sp
 293 .LP
 294 The `\fBddi_dmae_getlim()` function fills in the `\fBDMA` limit structure,
 295 pointed to by `\fIlimitsp`, with the `\fBDMA` limits of the system `\fBDMA`
 296 engine. Drivers for devices that perform their own bus mastering or use
 297 first-party `\fBDMA` must create and initialize their own `\fBDMA` limit
 298 structures; they should not use `\fBddi_dmae_getlim()`. The `\fBDMA` limit
 299 structure must be passed to the `\fBDMA` setup routines so that they will know
 300 how to break the `\fBDMA` request into windows. If the device has any

317 particular restrictions on transfer size or granularity (such as the size of
 318 disk sector), the driver should further restrict the values in the structure
 319 members before passing them to the `\fBDMA` setup routines. The driver must
 320 not relax any of the restrictions embodied in the structure after it is filled
 321 in by `\fBddi_dmae_getlim()`. After calling `\fBddi_dmae_getlim()`, a driver
 322 must examine, and possibly set, the size of the `\fBDMA` engine's
 323 `scatter/gather` list to determine whether `\fBDMA` chaining will be used. See
 324 `\fBddi_dma_lim_x86` and `\fBddi_dmae_req` for additional information
 325 on `scatter/gather` DMA.
 326 .SS "`\fBddi_dmae_getattr()`"
 327 .sp
 328 .LP
 329 The `\fBddi_dmae_getattr()` function fills in the `\fBDMA` attribute
 330 structure, pointed to by `\fIattrp`, with the `\fBDMA` attributes of the
 331 system `\fBDMA` engine. Drivers for devices that perform their own bus
 332 mastering or use first-party `\fBDMA` must create and initialize their own
 333 `\fBDMA` attribute structures; they should not use `\fBddi_dmae_getattr()`.
 334 The `\fBDMA` attribute structure must be passed to the `\fBDMA` resource
 335 allocation functions to provide the information necessary to break the
 336 `\fBDMA` request into `\fBDMA` windows and `\fBDMA` cookies. See
 337 `\fBddi_dma_nextcookie` and `\fBddi_dma_getwin` for additional information.
 338 .SH RETURN VALUES
 339 .sp
 340 .ne 2
 341 .na
 342 `\fB\fBDDI_SUCCESS`
 343 .ad
 344 .RS 23n
 345 Upon success, for all of these routines.
 346 .RE
 347 .sp
 348 .ne 2
 349 .na
 350 `\fB\fBDDI_FAILURE`
 351 .ad
 352 .RS 23n
 353 May be returned due to invalid arguments.
 354 .RE
 355 .sp
 356 .ne 2
 357 .na
 358 `\fB\fBDDI_DMA_NORESOURCES`
 359 .ad
 360 .RS 23n
 361 May be returned by `\fBddi_dmae_alloc()` if the requested resources are not
 362 available and the value of `\fIdmae_waitfp` is not `\fBDDI_DMA_SLEEP`.
 363 .RE
 364 .SH CONTEXT
 365 .sp
 366 .LP
 367 If `\fBddi_dmae_alloc()` is called from interrupt context, then its
 368 `\fIdmae_waitfp` argument and the callback function must not have the value
 369 `\fBDDI_DMA_SLEEP`. Otherwise, all these routines can be called from user,
 370 interrupt, or kernel context.
 371 .SH ATTRIBUTES
 372 .sp
 373 .LP
 374 See `\fBattributes` for descriptions of the following attributes:
 375 .sp
 376 .sp
 377 .TS
 378 box;

```
344 c | c
345 l | l .
346 ATTRIBUTE TYPE ATTRIBUTE VALUE
347 -
348 Architecture x86
349 .TE

351 .SH SEE ALSO
352 .sp
353 .LP
354 \fBisa\fR(4), \fBattributes\fR(5), \fBddi_dma_buf_setup\fR(9F),
355 \fBddi_dma_getwin\fR(9F), \fBddi_dma_nextcookie\fR(9F),
356 \fBddi_dma_mem_alloc\fR(9F), \fBddi_dma_addr_bind_handle\fR(9F), \fBddi_dma_attr
357 \fBddi_dma_cookie\fR(9S),
358 \fBddi_dma_cookie\fR(9S), \fBddi_dma_lim_x86\fR(9S), \fBddi_dma_req\fR(9S),
359 \fBddi_dmae_req\fR(9S)
```

new/usr/src/man/man9f/get_pktiopb.9f

1

```
*****
5136 Sat May 24 17:48:26 2014
new/usr/src/man/man9f/get_pktiopb.9f
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 \" te
2 .\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
3 .\" Copyright (c) 2006, Sun Microsystems, Inc., All Rights Reserved
4 .\" The contents of this file are subject to the terms of the Common Development
5 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
6 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
7 .TH GET_PKTIOPB 9F \"May 24, 2014\"
8 .TH GET_PKTIOPB 9F \"Jan 16, 2006\"
9 .SH NAME
10 get_pktiopb, free_pktiopb \- allocate/free a SCSI packet in the iopb map
11 .LP
12 .nf
13 #include <sys/scsi/scsi.h>

17 \fBstruct scsi_pkt *\fR\fBget_pktiopb\fR(\fBstruct scsi_address *\fR\fIap\fR,
18 \fBcaddr_t *\fR\fIidatap\fR, \fBint\fR \fIicdblen\fR, \fBint\fR \fIistatuslen\fR,
19 \fBint\fR \fIireadflag\fR, \fBint (*\fR\fIcallback\fR);
20 .fi

22 .LP
23 .nf
24 \fBvoid\fR \fBfree_pktiopb\fR(\fBstruct scsi_pkt *\fR\fIpkt\fR, \fBcaddr_t\fR \fI
25 .fi

27 .SH INTERFACE LEVEL
28 .sp
29 .LP
30 These interfaces are obsolete. Use \fBscsi_alloc_consistent_buf\fR(9F) instead
31 of \fBget_pktiopb()\fR. Use \fBscsi_free_consistent_buf\fR(9F) instead of
32 \fBfree_pktiopb()\fR.
33 .SH PARAMETERS
34 .sp
35 .ne 2
36 .na
37 \fB\fIap\fR\fR
38 .ad
39 .RS 13n
40 Pointer to the target's \fBscsi_address\fR structure.
41 .RE

43 .sp
44 .ne 2
45 .na
46 \fB\fIidatap\fR\fR
47 .ad
48 .RS 13n
49 Pointer to the address of the packet, set by this function.
50 .RE

52 .sp
53 .ne 2
54 .na
55 \fB\fIcdblen\fR\fR
```

new/usr/src/man/man9f/get_pktiopb.9f

2

```
56 .ad
57 .RS 13n
58 Number of bytes required for the \fBSCSI \fRcommand descriptor block (CDB).
59 .RE

61 .sp
62 .ne 2
63 .na
64 \fB\fIistatuslen\fR\fR
65 .ad
66 .RS 13n
67 Number of bytes required for the \fBSCSI \fRstatus area.
68 .RE

70 .sp
71 .ne 2
72 .na
73 \fB\fIidatalen\fR\fR
74 .ad
75 .RS 13n
76 Number of bytes required for the data area of the \fBSCSI \fRcommand.
77 .RE

79 .sp
80 .ne 2
81 .na
82 \fB\fIreadflag\fR\fR
83 .ad
84 .RS 13n
85 If non-zero, data will be transferred from the \fBSCSI \fRtarget.
86 .RE

88 .sp
89 .ne 2
90 .na
91 \fB\fIcallback\fR\fR
92 .ad
93 .RS 13n
94 Pointer to a callback function, or \fBNULL_FUNC\fR or \fBSLEEP_FUNC\fR
95 .RE

97 .sp
98 .ne 2
99 .na
100 \fB\fIpkt\fR\fR
101 .ad
102 .RS 13n
103 Pointer to a \fBscsi_pkt\fR(9S) structure.
104 .RE

106 .SH DESCRIPTION
107 .sp
108 .LP
109 The \fBget_pktiopb()\fR function allocates a \fBscsi_pkt\fR structure that has
110 a small data area allocated. It is used by some \fBSCSI \fRcommands such as
111 \fBREQUEST_SENSE\fR, which involve a small amount of data and require
112 cache-consistent memory for proper operation. It uses \fBddi_iopb_alloc\fR(9F)
113 for allocating the data area and \fBscsi_realloc\fR(9F) to allocate the packet
114 and \fBDMA\fR resources.
115 .sp
116 .LP
117 \fIcallback\fR indicates what \fBget_pktiopb()\fR should do when resources are
118 not available:
119 .sp
120 .ne 2
121 .na
```

```

122 \fB\FBNULL_FUNC\fR\fR
123 .ad
124 .RS 16n
125 Do not wait for resources. Return a \fINULL\fR pointer.
126 .RE

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

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

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

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

187 .SH SEE ALSO

```

```

188 .sp
189 .LP
190 \fBattributes\fR(5),
191 \fBattributes\fR(5), \fBddi_iopb_alloc\fR(9F),
192 \fBscsi_alloc_consistent_buf\fR(9F), \fBscsi_free_consistent_buf\fR(9F),
193 \fBscsi_pktalloc\fR(9F), \fBscsi_realloc\fR(9F), \fBscsi_pkt\fR(9S)
194 .LP
195 \fIWriting Device Drivers\fR
196 .SH NOTES
197 .sp
198 .LP
199 The \fBget_pktiopb()\fR and \fBfree_pktiopb()\fR functions are obsolete and
200 will be discontinued in a future release. These functions have been replaced
201 by, respectively, \fBscsi_alloc_consistent_buf\fR(9F) and
202 \fBscsi_free_consistent_buf\fR(9F).
203 .sp
204 .LP
205 The \fBget_pktiopb()\fR function uses scarce resources. For this reason and its
206 obsolescence (see above), its use is discouraged.

```



```

*****
7146 Sat May 24 17:48:26 2014
new/usr/src/man/man9f/scsi_hba_attach_setup.9f
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 \" te
2 .\" Copyright (c) 2006 Sun Microsystems, Inc., All Rights Reserved
3 .\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
4 .\" The contents of this file are subject to the terms of the Common Development
5 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
6 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
7 .TH SCSI_HBA_ATTACH_SETUP_9F \"May 24, 2014\"
8 .TH SCSI_HBA_ATTACH_SETUP_9F \"May 30, 2006\"
9 .SH NAME
10 scsi_hba_attach_setup, scsi_hba_detach \- SCSI HBA attach and
11 detach routines
12 .SH SYNOPSIS
13 .LP
14 #include <sys/scsi/scsi.h>

18 \fBint\fR \fBscsi_hba_attach_setup\fR(\fBdev_info_t * \fR\fR\fR, \fBddi_dma_at
19 \fBscsi_hba_tran_t * \fR\fR, \fBint\fR \fR, \fBflags\fR);
20 .fi

22 .LP
23 .nf
24 \fBint\fR \fBscsi_hba_attach\fR(\fBdev_info_t * \fR\fR, \fBddi_dma_lim_t * \fR
25 \fBscsi_hba_tran_t * \fR, \fBint\fR \fR, \fBflags\fR, \fBvoid * \fR
26 .fi

27 .LP
28 .nf
29 \fBint\fR \fBscsi_hba_detach\fR(\fBdev_info_t * \fR\fR);
30 .fi

27 .SH INTERFACE LEVEL
28 .sp
29 .LP
30 Solaris architecture specific (Solaris DDI).
31 .SH PARAMETERS
32 .sp
33 .ne 2
34 .na
35 \fB\fR
36 .ad
37 .RS 16n
38 Pointer to the \fBdev_info_t\fR structure that refers to the instance of the
39 HBA device.
40 .RE

42 .sp
43 .ne 2
44 .na
45 \fB\fR
46 .ad
47 .RS 16n
48 Pointer to a \fBddi_dma_lim\fR(9S) structure.

```

```

54 .RE

56 .sp
57 .ne 2
58 .na
59 \fB\fR
60 .ad
61 .RS 16n
62 Pointer to a \fBscsi_hba_tran\fR(9S) structure.
63 .RE

65 .sp
66 .ne 2
67 .na
68 \fB\fR
69 .ad
70 .RS 16n
71 Flag modifiers. The defined flag values are \fBSCSI_HBA_TRAN_CLONE\fR,
72 \fBSCSI_HBA_TRAN_SCB\fR, and \fBSCSI_HBA_TRAN_CDB\fR.
73 .RE

75 .sp
76 .ne 2
77 .na
78 \fB\fR
79 .ad
80 .RS 16n
81 Optional features provided by the HBA driver for future extensions; must be
82 \fB\fR.
83 .RE

85 .sp
86 .ne 2
87 .na
88 \fB\fR
89 .ad
90 .RS 16n
91 Pointer to a \fBddi_dma_attr\fR(9S) structure.
92 .RE

94 .SH DESCRIPTION
95 .sp
96 .SS "scsi_hba_attach_setup(\fR)"
97 .LP
98 The \fBscsi_hba_attach_setup(\fR) function is the recommended interface over
99 the \fBscsi_hba_attach(\fR) function.
100 .SS "scsi_hba_attach_setup(\fR) scsi_hba_attach(\fR)"
101 .sp
102 .LP
103 The \fBscsi_hba_attach_setup(\fR) function registers the
104 \fB\fR DMA limits and
105 the \fB\fR transport vectors of each instance of the HBA device defined
106 by \fB\fR. The \fBscsi_hba_attach_setup(\fR) function registers the
107 \fB\fR DMA attributes and the \fB\fR transport vectors of
108 each instance of the HBA device defined by \fB\fR. The HBA driver can pass
109 different DMA attributes and the transport vectors for each
110 instance of the device to support any constraints imposed by the HBA itself.
111 .sp
112 .LP
113 The \fBscsi_hba_attach_setup(\fR) function uses the
114 \fBscsi_hba_attach(\fR) and \fBscsi_hba_attach_setup(\fR) functions use the
115 \fBdev_bus_ops\fR field in the \fBdev_ops\fR(9S) structure. The HBA driver
116 should initialize this field to \fB\fR before calling
117 \fBscsi_hba_attach_setup(\fR).
118 \fBscsi_hba_attach(\fR) or \fBscsi_hba_attach_setup(\fR).

```

```

96 .sp
97 .LP
98 If \fBSCSI_HBA_TRAN_CLONE\fR is requested in \fIhba_flags\fR, the
99 \fBhba_tran\fR structure is cloned once for each target that is attached to the
100 HBA. The structure is cloned before the \fBtran_tgt_init\fR(9E) entry point is
101 called to initialize a target. At all subsequent HBA entry points, including
102 \fBtran_tgt_init\fR(9E), the \fBscsi_hba_tran_t\fR structure passed as an
103 argument or found in a \fBscsi_address\fR structure is the cloned
104 \fBscsi_hba_tran_t\fR structure, which allows the HBA to use the
105 \fBtran_tgt_private\fR field in the \fBscsi_hba_tran_t\fR structure to point to
106 per-target data. The HBA should free only the same \fBscsi_hba_tran_t\fR
107 structure allocated when the HBA detaches. All cloned \fBscsi_hba_tran_t\fR
108 structures that are allocated by the system are freed by the system.
109 .sp
110 .LP
111 The flags \fBSCSI_HBA_TRAN_CDB\fR and \fBSCSI_HBA_TRAN_SCB\fR are only valid
112 when \fBtran_setup_pkt()\fR is used. See \fBtran_setup_pkt\fR(9E) for
113 information on using these flags.
114 .sp
115 .LP
116 The \fBscsi_hba_attach_setup()\fR function attaches
117 The \fBscsi_hba_attach()\fR and \fBscsi_hba_attach_setup()\fR functions attach
118 a number of integer-valued properties to \fBfdip\fR, unless properties of the
119 same name are already attached to the node. An HBA driver should retrieve these
120 configuration parameters via \fBbdi_prop_get_int\fR(9F), and respect any
121 settings for features provided the HBA.
122 .ne 2
123 .na
124 \fB\fBscsi-options\fR\fR
125 .ad
126 .RS 26n
127 \fBOptional\fR \fBSCSI\fR \fBconfiguration bits\fR
128 .RE

130 .sp
131 .ne 2
132 .na
133 \fB\fBSCSI_OPTIONS_DR\fR\fR
134 .ad
135 .RS 26n
136 If not set, the HBA should not grant Disconnect privileges to target devices.
137 .RE

139 .sp
140 .ne 2
141 .na
142 \fB\fBSCSI_OPTIONS_TAG\fR\fR
143 .ad
144 .RS 26n
145 If not set, the HBA should not operate in Command Tagged Queueing mode.
146 .RE

148 .sp
149 .ne 2
150 .na
151 \fB\fBSCSI_OPTIONS_PARITY\fR\fR
152 .ad
153 .RS 26n
154 If not set, the HBA should not operate in parity mode.
155 .RE

157 .sp
158 .ne 2
159 .na
160 \fB\fBSCSI_OPTIONS_QAS\fR\fR

```

```

161 .ad
162 .RS 26n
163 If not set, the HBA should not make use of the Quick Arbitration Select
164 feature. Consult your Sun hardware documentation to determine whether your
165 machine supports QAS.
166 .RE

168 .sp
169 .ne 2
170 .na
171 \fB\fBSCSI_OPTIONS_FAST\fR\fR
172 .ad
173 .RS 26n
174 If not set, the HBA should not operate the bus in FAST SCSI mode.
175 .RE

177 .sp
178 .ne 2
179 .na
180 \fB\fBSCSI_OPTIONS_FAST20\fR\fR
181 .ad
182 .RS 26n
183 If not set, the HBA should not operate the bus in FAST20 SCSI mode.
184 .RE

186 .sp
187 .ne 2
188 .na
189 \fB\fBSCSI_OPTIONS_FAST40\fR\fR
190 .ad
191 .RS 26n
192 If not set, the HBA should not operate the bus in FAST40 SCSI mode.
193 .RE

195 .sp
196 .ne 2
197 .na
198 \fB\fBSCSI_OPTIONS_FAST80\fR\fR
199 .ad
200 .RS 26n
201 If not set, the HBA should not operate the bus in FAST80 SCSI mode.
202 .RE

204 .sp
205 .ne 2
206 .na
207 \fB\fBSCSI_OPTIONS_FAST160\fR\fR
208 .ad
209 .RS 26n
210 If not set, the HBA should not operate the bus in FAST160 SCSI mode.
211 .RE

213 .sp
214 .ne 2
215 .na
216 \fB\fBSCSI_OPTIONS_FAST320\fR\fR
217 .ad
218 .RS 26n
219 If not set, the HBA should not operate the bus in FAST320 SCSI mode.
220 .RE

222 .sp
223 .ne 2
224 .na
225 \fB\fBSCSI_OPTIONS_WIDE\fR\fR
226 .ad

```

```

227 .RS 26n
228 If not set, the HBA should not operate the bus in WIDE SCSI mode.
229 .RE

231 .sp
232 .ne 2
233 .na
234 \fB\fBSCSI_OPTIONS_SYNC\fR\fR
235 .ad
236 .RS 26n
237 If not set, the HBA should not operate the bus in synchronous transfer mode.
238 .RE

240 .sp
241 .ne 2
242 .na
243 \fB\fBscsi-reset-delay\fR\fR
244 .ad
245 .RS 26n
246 SCSI bus or device reset recovery time, in milliseconds.
247 .RE

249 .sp
250 .ne 2
251 .na
252 \fB\fBscsi-selection-timeout\fR\fR
253 .ad
254 .RS 26n
255 Default SCSI selection phase timeout value, in milliseconds. Please refer to
256 individual HBA man pages for any HBA-specific information
257 .RE

259 .SS "scsi_hba_detach(\|)"
260 .sp
261 .LP
262 The \fBscsi_hba_detach()\fR function removes the reference to the DMA
263 The \fBscsi_hba_detach()\fR function removes the reference to the DMA limits or
264 attributes structure and the transport vector for the given instance of an HBA
265 driver.
266 .SH RETURN VALUES
267 .sp
268 The \fBscsi_hba_attach_setup()\fR and
269 The \fBscsi_hba_attach()\fR, \fBscsi_hba_attach_setup()\fR, and
270 \fBscsi_hba_detach()\fR functions return \fBBDDI_SUCCESS\fR if the function call
271 succeeds, and return \fBBDDI_FAILURE\fR on failure.
272 .SH CONTEXT
273 .sp
274 The \fBscsi_hba_attach_setup()\fR function should
275 The \fBscsi_hba_attach()\fR and \fBscsi_hba_attach_setup()\fR functions should
276 be called from \fBattach\fR(9E). The \fBscsi_hba_detach()\fR function should be
277 called from \fBdetach\fR(9E).
278 .SH SEE ALSO
279 .sp
280 \fBattach\fR(9E), \fBdetach\fR(9E), \fBtrun_setup_pkt\fR(9E),
281 \fBtrun_tgt_init\fR(9E), \fBbdi_prop_get_int\fR(9F), \fBbdi_dma_attr\fR(9S),
282 \fBdev_ops\fR(9S), \fBscsi_address\fR(9S),
283 \fBbdi_dma_lim\fR(9S), \fBdev_ops\fR(9S), \fBscsi_address\fR(9S),
284 \fBscsi_hba_tran\fR(9S)
285 .LP
286 \fIWriting Device Drivers\fR
287 .SH NOTES
288 .sp

```

```

289 .LP
290 It is the HBA driver's responsibility to ensure that no more transport requests
291 will be taken on behalf of any SCSI target device driver after
292 \fBscsi_hba_detach()\fR is called.
293 .sp
294 The \fBscsi_hba_attach()\fR function is obsolete and will be discontinued in a
295 future release. This function is replaced by \fBscsi_hba_attach_setup()\fR.

```

```

*****
10572 Sat May 24 17:48:26 2014
new/usr/src/man/man9f/scsi_ifgetcap.9f
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 \" te
2 .\" Copyright (c) 2007, Sun Microsystems, Inc., All Rights Reserved
3 .\" Copyright (c) 2014 Garrett D'Amore <garrett@damore.org>
4 .\" The contents of this file are subject to the terms of the Common Development
5 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
6 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
7 .TH SCSI_IFGETCAP 9F \"May 24, 2014\"
8 .TH SCSI_IFGETCAP 9F \"Oct 16, 2007\"
9 .SH NAME
10 scsi_ifgetcap, scsi_ifsetcap \- get/set SCSI transport capability
11 .SH SYNOPSIS
12 .LP
13 #include <sys/scsi/scsi.h>

17 \fBint\fR \fBscsi_ifgetcap\fR(\fBstruct scsi_address *\fR\fIiap\fR, \fBchar *\fR\
18 .fi

20 .LP
21 .nf
22 \fBint\fR \fBscsi_ifsetcap\fR(\fBstruct scsi_address *\fR\fIiap\fR, \fBchar *\fR\
23 \fBint\fR \fIiwhom\fR);
24 .fi

26 .SH INTERFACE LEVEL
27 .sp
28 .LP
29 Solaris DDI specific (Solaris DDI).
30 .SH PARAMETERS
31 .sp
32 .ne 2
33 .na
34 \fB\fIiap\fR\fR
35 .ad
36 .RS 9n
37 Pointer to the \fBscsi_address\fR structure.
38 .RE

40 .sp
41 .ne 2
42 .na
43 \fB\fIicap\fR\fR
44 .ad
45 .RS 9n
46 Pointer to the string capability identifier.
47 .RE

49 .sp
50 .ne 2
51 .na
52 \fB\fIvalue\fR\fR
53 .ad
54 .RS 9n
55 Defines the new state of the capability.

```

```

56 .RE

58 .sp
59 .ne 2
60 .na
61 \fB\fIwhom\fR\fR
62 .ad
63 .RS 9n
64 Determines if all targets or only the specified target is affected.
65 .RE

67 .SH DESCRIPTION
68 .sp
69 .LP
70 The \fBscsi_ifsetcap()\fR function is used by target drivers to set the
71 capabilities of the host adapter driver. The \fIicap\fR pointer is a name-value
72 pair identified by a null-terminated character string and the integer value of
73 the \fIicap\fR. The current value of the capability can be retrieved with the
74 \fBscsi_ifgetcap()\fR function. If the \fIiwhom\fR value is \fB0\fR, all target
75 drivers are affected. Otherwise, the \fBscsi_address\fR structure pointed to by
76 \fIiap\fR is the only target that is affected.
77 .sp
78 .LP
79 The driver should confirm that \fBscsi_ifsetcap()\fR and \fBscsi_ifsetcap()\fR
80 functions are called with a \fIicap\fR that points to a capability which is
81 supported by the device.
82 .sp
83 .LP
84 The following capabilities have been defined:
85 .sp
86 .ne 2
87 .na
88 \fB\fBdma-max\fR\fR
89 .ad
90 .RS 24n
91 Maximum \fBdma\fR transfer size that is supported by the host adapter.
92 .RE

94 .sp
95 .ne 2
96 .na
97 \fB\fBdma-max-arch\fR\fR
98 .ad
99 .RS 24n
100 Maximum \fBdma\fR transfer size that is supported by system. Takes the host
101 adapter and system architecture into account. This is useful for target drivers
102 which do not support partial \fBDMA\fRs on systems which do not have an
103 \fBBIOMMU\fR. In this case, the \fBDMA\fR can also be limited by the host
104 adapters "scatter/gather" list constraints.
105 .sp
106 The "\fBdma-max-arch\fR" capability can not be set. It is implemented with this
107 command and does not rely on a \fBtran_getcap\fR(9E) response from the HBA.
108 .RE

110 .sp
111 .ne 2
112 .na
113 \fB\fBmsg-out\fR\fR
114 .ad
115 .RS 24n
116 Message out capability that is supported by the host adapter: \fB0\fR disables,
117 \fB1\fR enables.
118 .RE

120 .sp
121 .ne 2

```

```

122 .na
123 \fB\fBdisconnect\fR\fR
124 .ad
125 .RS 24n
126 Disconnect capability that is supported by the host adapter: \fB0\fR disables,
127 \fB1\fR enables.
128 .RE

130 .sp
131 .ne 2
132 .na
133 \fB\fB asynchronous\fR\fR
134 .ad
135 .RS 24n
136 Synchronous data transfer capability that is supported by the host adapter:
137 \fB0\fR disables, \fB1\fR enables.
138 .RE

140 .sp
141 .ne 2
142 .na
143 \fB\fB wide-xfer\fR\fR
144 .ad
145 .RS 24n
146 Wide transfer capability that is supported by the host adapter: \fB0\fR
147 disables, \fB1\fR enables.
148 .RE

150 .sp
151 .ne 2
152 .na
153 \fB\fB parity\fR\fR
154 .ad
155 .RS 24n
156 Parity checking capability that is supported by host adapter: \fB0\fR disables,
157 \fB1\fR enables.
158 .RE

160 .sp
161 .ne 2
162 .na
163 \fB\fB initiator-id\fR\fR
164 .ad
165 .RS 24n
166 Host bus address that is returned.
167 .RE

169 .sp
170 .ne 2
171 .na
172 \fB\fB untagged-qing\fR\fR
173 .ad
174 .RS 24n
175 Host adapter capability that supports internal queuing of commands without
176 tagged queuing: \fB0\fR disables, \fB1\fR enables.
177 .RE

179 .sp
180 .ne 2
181 .na
182 \fB\fB tagged-qing\fR\fR
183 .ad
184 .RS 24n
185 Host adapter capability that supports queuing: \fB0\fR disables, \fB1\fR
186 enables.
187 .RE

```

```

189 .sp
190 .ne 2
191 .na
192 \fB\fB auto-rqsense\fR\fR
193 .ad
194 .RS 24n
195 Host adapter capability that supports auto request sense on check conditions:
196 \fB0\fR disables, \fB1\fR enables.
197 .RE

199 .sp
200 .ne 2
201 .na
202 \fB\fB sector-size\fR\fR
203 .ad
204 .RS 24n
205 Capability that is set by the target driver to inform the \fBHBA\fR of the
206 granularity, in bytes, of the \fBDMA\fR breakup. The \fBHBA\fR \fBDMA\fR
207 attributes structure is set to reflect the byte total of this setting. See
208 \fBddi_dma_attribut\fR(9S). The \fBsector-size\fR
209 granularity, in bytes, of the \fBDMA\fR breakup. The \fBHBA\fR \fBDMA\fR limit
210 structure is set to reflect the byte total of this setting. See
211 \fBddi_dma_lim_sparc\fR(9S) or \fBddi_dma_lim_x86\fR(9S). The \fBsector-size\fR
212 should be set to the size of the physical disk sector. The capability defaults
213 to 512 bytes.
214 .RE

213 .sp
214 .ne 2
215 .na
216 \fB\fB total-sectors\fR\fR
217 .ad
218 .RS 24n
219 Capability that is set by the target driver to inform the \fBHBA\fR of the
220 total number of sectors on the device returned by the \fBSCSI\fR \fBget
221 capacity\fR command. This capability must be set before the target driver
222 'gets' the \fBgeometry\fR capability.
223 .RE

225 .sp
226 .ne 2
227 .na
228 \fB\fB geometry\fR\fR
229 .ad
230 .RS 24n
231 Capability that returns the \fBHBA\fR geometry of a target disk. The target
232 driver sets the \fBtotal-sectors\fR capability before 'getting' the geometry
233 capability. The geometry is returned as a 32-bit value. The upper 16 bits
234 represent the number of heads per cylinder. The lower 16 bits represent the
235 number of sectors per track. The geometry capability cannot be 'set'.
236 .sp
237 If geometry is not relevant or appropriate for the target disk,
238 \fBscsi_ifgetcap()\fR can return \fB-1\fR to indicate that the geometry is not
239 defined. For example, if the \fBHBA\fR BIOS supports Logical Block Addressing
240 for the target disk, \fBscsi_ifgetcap()\fR returns \fB-1\fR. Attempts to
241 retrieve the "virtual geometry" from the target driver, such as the
242 \fBBDKIOCG_VIRTGEOM\fR ioctl, will fail. See \fBdkio\fR(7I) for more information
243 about \fBBDKIOCG_VIRTGEOM\fR.
244 .RE

246 .sp
247 .ne 2
248 .na
249 \fB\fB reset-notification\fR\fR
250 .ad

```

```

251 .RS 24n
252 Host adapter capability that supports bus reset notification: \fB0\fR disables,
253 \fB1\fR enables. See \fBscsi_reset_notify\fR(9F).
254 .RE

256 .sp
257 .ne 2
258 .na
259 \fB\bblinked-cmds\fR\fR
260 .ad
261 .RS 24n
262 Host adapter capability that supports linked commands: \fB0\fR disables,
263 \fB1\fR enables.
264 .RE

266 .sp
267 .ne 2
268 .na
269 \fB\bqfull-retries\fR\fR
270 .ad
271 .RS 24n
272 Capability that enables or disables \fBQUEUE\fR \fBFULL\fR handling. If
273 \fB0\fR, the \fBHBA\fR will not retry a command when a \fBQUEUE\fR \fBFULL\fR
274 status is returned. If the value is greater than \fB0\fR, the \fBHBA\fR driver
275 retries the command a specified number of times at an interval determined by
276 the \fBqfull-retry-interval\fR. The range for \fBqfull-retries\fR is
277 \fB0-255\fR.
278 .RE

280 .sp
281 .ne 2
282 .na
283 \fB\bqfull-retry-interval\fR\fR
284 .ad
285 .RS 24n
286 Capability that sets the retry interval in milliseconds (\fBms\fR) for commands
287 completed with a \fBQUEUE\fR \fBFULL\fR status. The range for
288 \fBqfull-retry-intervals\fR is \fB0-1000\fR \fBms\fR.
289 .RE

291 .sp
292 .ne 2
293 .na
294 \fB\bblun-reset\fR\fR
295 .ad
296 .RS 24n
297 Capability that is created with a value of zero by \fBHBA\fR drivers that
298 support the \fBRESET_LUN\fR flag in the \fBtrun_reset\fR(9E) function. If it
299 exists, the \fBlun-reset\fR value can be set to \fB1\fR by target drivers to
300 allow the use of \fBLOGICAL UNIT RESET\fR on a specific target instance. If
301 \fBlun-reset\fR does not exist or has a value of zero, \fBscsi_reset\fR(9F) is
302 prevented from passing the \fBRESET_LUN\fR flag to \fBtrun_reset()\fR function
303 of the \fBHBA\fR driver. If \fBlun-reset\fR exists and has a value of \fB1\fR,
304 the \fBtrun_reset()\fR function of the \fBHBA\fR driver can be called with the
305 \fBRESET_LUN\fR flag.
306 .RE

308 .sp
309 .ne 2
310 .na
311 \fB\binterconnect-type\fR
312 .ad
313 .RS 24n
314 Capability held in the \fBtrun_interconnect_type\fR element of struct
315 \fBscsi_hba_tran\fR that indicates the \fBHBA\fR transport interconnect type .
316 The integer value of the interconnect type of the transport is defined in the

```

```

317 \fBservices.h\fR header file.
318 .RE

320 .sp
321 .ne 2
322 .na
323 \fB\bmax-cdb-length\fR
324 .ad
325 .RS 24n
326 Host adapter capability of the maximum supported \fBCDB\fR (Command Descriptor
327 Block) length. The target driver asks for the capability at attach time. If the
328 \fBHBA\fR driver supports the capability, the maximum length of the \fBCDB\fR
329 is returned in bytes. The target driver can then use that value to determine
330 which \fBCDB\fR is used for the \fBHBA\fR.
331 .sp
332 If the \fBHBA\fR driver does not support the \fBmax-cdb-length\fR capability,
333 the default value of the target driver is used for the \fBCDB\fR determination.
334 .RE

336 .SH RETURN VALUES
337 .sp
338 .LP
339 The \fBscsi_ifsetcap()\fR function returns:
340 .sp
341 .ne 2
342 .na
343 \fB\b1\fR\fR
344 .ad
345 .RS 9n
346 If the capability was successfully set to the new value.
347 .RE

349 .sp
350 .ne 2
351 .na
352 \fB\b0\fR\fR
353 .ad
354 .RS 9n
355 If the capability is not variable.
356 .RE

358 .sp
359 .ne 2
360 .na
361 \fB\b(mil)\fR\fR
362 .ad
363 .RS 9n
364 If the capability was not defined, or setting the capability to a new value
365 failed.
366 .RE

368 .sp
369 .LP
370 The \fBscsi_ifgetcap()\fR function returns the current value of a capability,
371 or:
372 .sp
373 .ne 2
374 .na
375 \fB\b(mil)\fR\fR
376 .ad
377 .RS 9n
378 If the capability was not defined.
379 .RE

381 .SH EXAMPLES
382 .LP

```

```
383 \fBExample 1 \fRUsing \fBscsi_ifgetcap()\fR
384 .sp
385 .in +2
386 .nf
387 if (scsi_ifgetcap(&sd->sd_address, "auto-rqsense", 1) == 1) {
388     un->un_arq_enabled = 1;
389 } else {
    unchanged_portion_omitted
405 .fi
406 .in -2

408 .SH CONTEXT
409 .sp
410 .LP
411 These functions can be called from user, interrupt, or kernel context.
412 .SH ATTRIBUTES
413 .sp
414 .LP
415 See \fBattributes\fR(5) for descriptions of the following attributes:
416 .sp

418 .sp
419 .TS
420 box;
421 c | c
422 l | l .
423 ATTRIBUTE TYPE    ATTRIBUTE VALUE
424 -
425 Interface Stability    Committed
426 .TE

428 .SH SEE ALSO
429 .sp
430 .LP
431 \fBbtran_reset\fR(9E), \fBscsi_hba_lookup_capstr\fR(9F), \fBscsi_reset\fR(9F),
432 \fBscsi_reset_notify\fR(9F), \fBddi_dma_attr\fR(9S),
433 \fBscsi_address\fR(9S), \fBscsi_arq_status\fR(9S)
434 \fBscsi_reset_notify\fR(9F), \fBddi_dma_lim_sparc\fR(9S),
435 \fBddi_dma_lim_x86\fR(9S), \fBscsi_address\fR(9S), \fBscsi_arq_status\fR(9S)
436 .sp
437 .LP
438 \fIWriting Device Drivers\fR
```

10322 Sat May 24 17:48:27 2014

new/usr/src/man/man9f/scsi_init_pkt.9f

4888 Undocument dma_req(9s)

4884 EOF scsi_hba_attach

4886 EOF ddi_dmae_getlim

4887 EOF ddi_iomin

4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)

4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free

```

1  \" te
2  .\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
3  .\" Copyright (c) 2006, Sun Microsystems, Inc., All Rights Reserved
4  .\" The contents of this file are subject to the terms of the Common Development
5  .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
6  .\" When distributing Covered Code, include this CDDL HEADER in each file and in
7  .TH SCSI_INIT_PKT 9F \"May 24, 2014\"
8  .TH SCSI_INIT_PKT 9F \"Jan 16, 2006\"
9  .SH NAME
10 scsi_init_pkt \- prepare a complete SCSI packet
11 .SH SYNOPSIS
12 .LP
13 #include <sys/scsi/scsi.h>

```

```

17 \fBstruct scsi_pkt *\fR\fBscsi_init_pkt\fR(\fBstruct scsi_address *\fR\fIiap\fR,
18 \fBstruct scsi_pkt *\fR\fIpktp\fR, \fBstruct buf *\fR\fIbp\fR, \fBint\fR \fI
19 \fBint\fR \fIprivatelen\fR, \fBint\fR \fIflags\fR, \fBint\fR \fIb*\fRcallba
20 .fi

```

```

22 .SH INTERFACE LEVEL
23 .sp
24 .LP
25 Solaris DDI specific (Solaris DDI).
26 .SH PARAMETERS
27 .sp
28 .ne 2
29 .na
30 \fBfiap\fR
31 .ad
32 .sp .6
33 .RS 4n
34 Pointer to a \fBscsi_address\fR(9S) structure.
35 .RE

```

```

37 .sp
38 .ne 2
39 .na
40 \fBipktp\fR
41 .ad
42 .sp .6
43 .RS 4n
44 A pointer to a \fBscsi_pkt\fR(9S) structure.
45 .RE

```

```

47 .sp
48 .ne 2
49 .na
50 \fBibp\fR
51 .ad
52 .sp .6
53 .RS 4n
54 Pointer to a \fBbuf\fR(9S) structure.
55 .RE

```

```

57 .sp
58 .ne 2
59 .na
60 \fBicmdlen\fR
61 .ad
62 .sp .6
63 .RS 4n
64 The required length for the \fBSCSI \fRcommand descriptor block (\fBCDB\fR) in
65 bytes.
66 .RE

```

```

68 .sp
69 .ne 2
70 .na
71 \fBistatuslen\fR
72 .ad
73 .sp .6
74 .RS 4n
75 The required length for the \fBSCSI \fRstatus completion block (\fBSCB\fR) in
76 bytes. Valid values are:
77 .sp
78 .ne 2
79 .na
80 \fBfb0\fR
81 .ad
82 .sp .6
83 .RS 4n
84 No status back.
85 .RE

```

```

87 .sp
88 .ne 2
89 .na
90 \fBfb1\fR
91 .ad
92 .sp .6
93 .RS 4n
94 Return SCSI status byte.
95 .RE

```

```

97 .sp
98 .ne 2
99 .na
100 \fBfbsizeof(scsi_arq_status)\fR
101 .ad
102 .sp .6
103 .RS 4n
104 Return status information in a \fBscsi_arq_status\fR structure. This will
105 include up to 20 bytes of sense data. Please refer to \fBscsi_arq_status\fR(9S)
106 for more information.
107 .sp
108 For extra sense packets (\fBPKT_XARQ\fR flag asserted), set \fIstatuslen\fR to
109 be a greater number like, (\fIN\fR + \fBsizeof(struct scsi_arq_status)\fR)
110 where \fIN\fR is the number of extra bytes beyond the default 20. For example,
111 \fIN\fR=1 requests 21 bytes of sense, \fIN\fR=235 asks for 255 bytes.
112 .RE

```

```
114 .RE
```

```

116 .sp
117 .ne 2
118 .na
119 \fBfprivatelen\fR
120 .ad
121 .sp .6

```



```

122 .RS 4n
123 The required length for the \fIpkt_private\fR area.
124 .RE

126 .sp
127 .ne 2
128 .na
129 \fB\fIflags\fR\fR
130 .ad
131 .sp .6
132 .RS 4n
133 Flags modifier.
134 .RE

136 .sp
137 .ne 2
138 .na
139 \fB\fIcallback\fR\fR
140 .ad
141 .sp .6
142 .RS 4n
143 A pointer to a callback function, \fBNULL_FUNC\fR, or \fBSLEEP_FUNC\fR.
144 .RE

146 .sp
147 .ne 2
148 .na
149 \fB\fIarg\fR\fR
150 .ad
151 .sp .6
152 .RS 4n
153 The \fIcallback\fR function argument.
154 .RE

156 .SH DESCRIPTION
157 .sp
158 .LP
159 Target drivers use \fBscsi_init_pkt()\fR to request the transport layer to
160 allocate and initialize a packet for a \fBSCSI\fR command which possibly
161 includes a data transfer. If \fIpkt\fR is \fINULL\fR, a new \fBscsi_pkt\fR(9S)
162 is allocated using the \fBHBA\fR driver's packet allocator. The \fIbp\fR is a
163 pointer to a \fBbuf\fR(9S) structure. If \fIbp\fR is non-\fINULL\fR and
164 contains a valid byte count, the \fBbuf\fR(9S) structure is also set up for
165 \fBDMA \fRtransfer using the \fBHBA\fR driver \fBDMA\fR resources allocator.
166 When \fIbp\fR is allocated by \fBscsi_alloc_consistent_buf\fR(9F), the
167 \fBPKT_CONSISTENT\fR bit must be set in the \fIflags\fR argument to ensure
168 proper operation. If \fIprivatelen\fR is non-zero then additional space is
169 allocated for the \fIpkt_private\fR area of the \fBscsi_pkt\fR(9S). On return
170 \fIpkt_private\fR points to this additional space. Otherwise \fIpkt_private\fR
171 is a pointer that is typically used to store the \fIbp\fR during execution of
172 the command. In this case \fIpkt_private\fR is \fINULL\fR on return.
173 .sp
174 .LP
175 The \fIflags\fR argument is a set of bit flags. Possible bits include:
176 .sp
177 .ne 2
178 .na
179 \fB\fBPKT_CONSISTENT\fR\fR
180 .ad
181 .sp .6
182 .RS 4n
183 This must be set if the \fBDMA\fR buffer was allocated using
184 \fBscsi_alloc_consistent_buf\fR(9F). In this case, the \fBHBA\fR driver will
185 guarantee that the data transfer is properly synchronized before performing the
186 target driver's command completion callback.
187 .RE

```

```

189 .sp
190 .ne 2
191 .na
192 \fB\fBPKT_DMA_PARTIAL\fR\fR
193 .ad
194 .sp .6
195 .RS 4n
196 This may be set if the driver can accept a partial \fBDMA\fR mapping. If set,
197 \fBscsi_init_pkt()\fR will allocate \fBDMA\fR resources with the
198 \fBDDI_DMA_PARTIAL\fR bit set in the DMA flags.
199 The \fBpkt_resid\fR field of the
200 \fBDDI_DMA_PARTIAL\fR bit set in the \fBdmr_flag\fR element of the
201 \fBddi_dma_req\fR(9S) structure. The \fBpkt_resid\fR field of the
202 \fBscsi_pkt\fR(9S) structure may be returned with a non-zero value, which
203 indicates the number of bytes for which \fBscsi_init_pkt()\fR was unable to
204 allocate DMA resources. In this case, a subsequent call to
205 \fBscsi_init_pkt()\fR may be made for the same \fIpkt\fR and \fIbp\fR to
206 adjust the DMA resources to the next portion of the transfer. This sequence
207 should be repeated until the \fBpkt_resid\fR field is returned with a zero
208 value, which indicates that with transport of this final portion the entire
209 original request will have been satisfied.
210 .RE

210 .sp
211 .ne 2
212 .na
213 \fB\fBPKT_XARQ\fR\fR
214 .ad
215 .sp .6
216 .RS 4n
217 Setting this flag requests that the \fBHBA\fR return extra sense data for this
218 \fBscsi_pkt\fR(9S). The default auto request sense mechanism returns up to 20
219 bytes. More than 20 bytes of sense data can be requested by setting this flag
220 and setting the \fIstatuslen\fR correctly. Set the \fIstatuslen\fR to be the
221 \fBsizeof(struct scsi_arq_status)\fR plus the number of sense bytes needed
222 beyond 20. For example, set statuslen to be \fB(sizeof(struct scsi_arq_status)
223 + 5)\fR for 25 bytes of sense.
224 .RE

226 .sp
227 .LP
228 When calling \fBscsi_init_pkt()\fR to move already-allocated \fBDMA\fR
229 resources, the \fIcmdlen\fR, \fIstatuslen\fR, and \fIprivatelen\fR fields are
230 ignored.
231 .sp
232 .LP
233 The last argument \fIarg\fR is supplied to the \fIcallback\fR function when it
234 is invoked.
235 .sp
236 .LP
237 \fIcallback\fR indicates what the allocator routines should do when resources
238 are not available:
239 .sp
240 .ne 2
241 .na
242 \fB\fBNULL_FUNC\fR\fR
243 .ad
244 .RS 16n
245 Do not wait for resources. Return a \fINULL\fR pointer.
246 .RE

248 .sp
249 .ne 2
250 .na
251 \fB\fBSLEEP_FUNC\fR\fR

```

```

252 .ad
253 .RS 16n
254 Wait indefinitely for resources.
255 .RE

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

271 .sp
272 .LP
273 When allocating \fBDMA\fR resources, \fBscsi_init_pkt()\fR returns the
274 \fBscsi_pkt\fR field \fBpkt_resid\fR as the number of residual bytes for which
275 the system was unable to allocate \fBDMA\fR resources. A \fBpkt_resid\fR of
276 \fB0\fR means that all necessary \fBDMA\fR resources were allocated.
277 .SH RETURN VALUES
278 .sp
279 .LP
280 The \fBscsi_init_pkt()\fR function returns \fBINULL\fR if the packet or
281 \fBDMA\fR resources could not be allocated. Otherwise, it returns a pointer to
282 an initialized \fBscsi_pkt\fR(9S). If \fBpkt\fR was not \fBINULL\fR the return
283 value will be \fBpkt\fR on successful initialization of the packet.
284 .SH CONTEXT
285 .sp
286 .LP
287 If \fIcallback\fR is \fBSLEEP_FUNC\fR, then this routine can be called only
288 from user-level code. Otherwise, it can be called from user, interrupt, or
289 kernel context. The \fIcallback\fR function may not block or call routines that
290 block.
291 .SH EXAMPLES
292 .LP
293 \fBExample 1\fR \fRAllocating a Packet Without \fBDMA\fR Resources Attached
294 .sp
295 .LP
296 To allocate a packet without \fBDMA\fR resources attached, use:

298 .sp
299 .in +2
300 .nf
301 pkt = scsi_init_pkt(&devp->sd_address, NULL, NULL, CDB_GROUP1,
302     1, sizeof (struct my_pkt_private *), 0,
303     sd_runout, sd_unit);
304 .fi
305 .in -2

307 .LP
308 \fBExample 2\fR \fRAllocating a Packet With \fBDMA\fR Resources Attached
309 .sp
310 .LP
311 To allocate a packet with \fBDMA\fR resources attached use:

313 .sp
314 .in +2
315 .nf
316 pkt = scsi_init_pkt(&devp->sd_address, NULL, bp, CDB_GROUP1,
317     sizeof(struct scsi_arg_status), 0, 0, NULL_FUNC, NULL);

```

```

318 .fi
319 .in -2

321 .LP
322 \fBExample 3\fR \fRAttaching \fBDMA\fR Resources to a Preallocated Packet
323 .sp
324 .LP
325 To attach \fBDMA\fR resources to a preallocated packet, use:

327 .sp
328 .in +2
329 .nf
330 pkt = scsi_init_pkt(&devp->sd_address, old_pkt, bp, 0,
331     0, 0, 0, sd_runout, (caddr_t) sd_unit);
332 .fi
333 .in -2

335 .LP
336 \fBExample 4\fR \fRAllocating a Packet with Consistent \fBDMA\fR Resources
337 Attached
338 .sp
339 .LP
340 Since the packet is already allocated, the \fIcmdlen\fR, \fIstatuslen\fR and
341 \fIprivate_len\fR are \fB0\fR. To allocate a packet with consistent \fBDMA\fR
342 resources attached, use:

344 .sp
345 .in +2
346 .nf
347 bp = scsi_alloc_consistent_buf(&devp->sd_address, NULL,
348     SENSE_LENGTH, B_READ, SLEEP_FUNC, NULL);
349     pkt = scsi_init_pkt(&devp->sd_address, NULL, bp, CDB_GROUP0,
350     sizeof(struct scsi_arg_status), sizeof (struct my_pkt_private *),
351     PKT_CONSISTENT, SLEEP_FUNC, NULL);
352 .fi
353 .in -2

355 .LP
356 \fBExample 5\fR \fRAllocating a Packet with Partial \fBDMA\fR Resources Attached
357 .sp
358 .LP
359 To allocate a packet with partial \fBDMA\fR resources attached, use:

361 .sp
362 .in +2
363 .nf
364 my_pkt = scsi_init_pkt(&devp->sd_address, NULL, bp, CDB_GROUP0,
365     1, sizeof (struct buf *), PKT_DMA_PARTIAL,
366     SLEEP_FUNC, NULL);
367 .fi
368 .in -2

370 .SH SEE ALSO
371 .sp
372 .LP
373 \fBscsi_alloc_consistent_buf\fR(9F), \fBscsi_destroy_pkt\fR(9F),
374 \fBscsi_dmaget\fR(9F), \fBscsi_pktalloc\fR(9F), \fBbbuf\fR(9S),
375 \fBscsi_address\fR(9S), \fBscsi_pkt\fR(9S)
376 .sp
377 .LP
378 \fIWriting Device Drivers\fR
379 .SH NOTES
380 .sp
381 .LP
382 If a \fBDMA\fR allocation request fails with \fBBDMA_NOMAPPING\fR, the

```

383 \fBB_ERROR\fR flag will be set in \fIbp\fR, and the \fBb_error\fR field will be
384 set to \fBDEFAULT\fR.
385 .sp
386 .LP
387 If a \fBDMA\fR allocation request fails with \fBDDI_DMA_TOO_BIG\fR, the
388 \fBB_ERROR\fR flag will be set in \fIbp\fR, and the \fBb_error\fR field will be
389 set to \fBINVAL\fR.

```

*****
4050 Sat May 24 17:48:27 2014
new/usr/src/man/man9s/Intro.9s
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 \" te
2 .\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
3 .\" Copyright (c) 2001, Sun Microsystems, Inc., All Rights Reserved.
4 .\" Copyright 1989 AT&T
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 9S \"May 24, 2014\"
7 .TH INTRO 9S \"May 15, 2001\"
9 .SH NAME
10 Intro, intro \- introduction to kernel data structures and properties
11 .SH DESCRIPTION
12 .sp
13 .LP
14 Section 9P describes kernel properties used by device drivers. Section 9S
15 describes the data structures used by drivers to share information between the
16 driver and the kernel. See \fBIntro\fR(9E) for an overview of device driver
17 interfaces.
18 .sp
19 .LP
20 In Section 9S, reference pages contain the following headings:
21 .RS +4
22 .TP
23 .ie t \(\bu
24 .el o
25 \fBNAME\fR summarizes the purpose of the structure or property.
26 .RE
27 .RS +4
28 .TP
29 .ie t \(\bu
30 .el o
31 \fBSYNOPSIS\fR lists the include file that defines the structure or property.
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 \fBDESCRIPTION\fR provides general information about the structure or property.
44 .RE
45 .RS +4
46 .TP
47 .ie t \(\bu
48 .el o
49 \fBSTRUCTURE\fR \fBMEMBERS\fR lists all accessible structure members (for
50 Section 9S).
51 .RE
52 .RS +4
53 .TP
54 .ie t \(\bu
55 .el o

```

```

56 \fBSEE\fR \fBALSO\fR gives sources for further information.
57 .RE
58 .sp
59 .LP
60 Of the preceding headings, Section 9P reference pages contain the \fBNAME\fR,
61 \fBDESCRIPTION\fR, and \fBSEE\fR \fBALSO\fR fields.
62 .sp
63 .LP
64 Every driver MUST include <\fBsys/ddi.h\fR> and <\fBsys/sunddi.h\fR>, in that
65 order, and as final entries.
66 .sp
67 .LP
68 The following table summarizes the STREAMS structures described in Section 9S.
69 .sp
71 .sp
72 .TS
73 box;
74 c | c
75 l | l .
76 Structure          Type
77 -
78 \fBcopyreq\fR      DDI/DKI
79 -
80 \fBcopyresp\fR     DDI/DKI
81 -
82 \fBdatab\fR        DDI/DKI
83 -
84 \fBfmodsw\fR       Solaris DDI
85 -
86 \fBfree_rtn\fR     DDI/DKI
87 -
88 \fBbiocblk\fR      DDI/DKI
89 -
90 \fBblinkblk\fR     DDI/DKI
91 -
92 \fBmodule_info\fR  DDI/DKI
93 -
94 \fBmsgb\fR         DDI/DKI
95 -
96 \fBqband\fR        DDI/DKI
97 -
98 \fBqinit\fR        DDI/DKI
99 -
100 \fBqueclass\fR     Solaris DDI
101 -
102 \fBqueue\fR        DDI/DKI
103 -
104 \fBstreamtab\fR   DDI/DKI
105 -
106 \fBstroptions\fR  DDI/DKI
107 .TE
109 .sp
110 .LP
111 The following table summarizes structures that are not specific to STREAMS I/O.
112 .sp
114 .sp
115 .TS
116 box;
117 c | c
118 l | l .
119 Structure          Type
120 -
121 \fBbaio_req\fR     Solaris DDI

```

```

122 _
123 \fBbuf\fR      DDI/DKI
124
125 \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 _
133 \fBddi_dma_lim_sparc\fR      Solaris SPARC DDI
134 \fBddi_dma_lim_x86\fR      Solaris x86 DDI
135 _
136 \fBddi_dma_req\fR      Solaris DDI
137
138 \fBddi_dmae_req\fR      Solaris x86 DDI
139
140 \fBddi_idevice_cookie\fR      Solaris DDI
141
142 \fBddi_mapdev_ctl\fR      Solaris DDI
143
144 \fBdevmap_callback_ctl\fR      Solaris DDI
145
146 \fBdev_ops\fR      Solaris DDI
147
148 \fBbiovec\fR      DDI/DKI
149
150 \fBbkstat\fR      Solaris DDI
151
152 \fBbkstat_intr\fR      Solaris DDI
153
154 \fBbkstat_io\fR      Solaris DDI
155
156 \fBbkstat_named\fR      Solaris DDI
157
158 \fBbmap\fR      DDI/DKI
159
160 \fBbmoddrv\fR      Solaris DDI
161
162 \fBbmodlinkage\fR      Solaris DDI
163
164 \fBbmodlstrmod\fR      Solaris DDI
165
166 \fBbscsi_address\fR      Solaris DDI
167
168 \fBbscsi_arq_status\fR      Solaris DDI
169
170 \fBbscsi_device\fR      Solaris DDI
171
172 \fBbscsi_extended_sense\fR      Solaris DDI
173
174 \fBbscsi_hba_tran\fR      Solaris DDI
175
176 \fBbscsi_inquiry\fR      Solaris DDI
177
178 \fBbscsi_pkt\fR      Solaris DDI
179
180 \fBbscsi_status\fR      Solaris DDI
181
182 \fBuio\fR      DDI/DKI
183
184 .TE
185
186 .SH SEE ALSO
187
188 .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.

```

new/usr/src/man/man9s/Makefile

1

2345 Sat May 24 17:48:27 2014

new/usr/src/man/man9s/Makefile

4888 Undocument dma_req(9s)

4884 EOF scsi_hba_attach

4886 EOF ddi_dmae_getlim

4887 EOF ddi_iomin

4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)

4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free

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

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

18 include \$(SRC)/Makefile.master

20 MANSECT= 9s

```
22 MANFILES= Intro.9s \
23 aio_req.9s \
24 buf.9s \
25 cb_ops.9s \
26 copyreq.9s \
27 copyresp.9s \
28 datab.9s \
29 ddi_device_acc_attr.9s \
30 ddi_dma_attr.9s \
31 ddi_dma_cookie.9s \
32 ddi_dma_lim_sparc.9s \
33 ddi_dma_lim_x86.9s \
34 ddi_dma_req.9s \
35 ddi_dmae_req.9s \
36 ddi_fm_error.9s \
37 ddi_idevice_cookie.9s \
38 dev_ops.9s \
39 devmap_callback_ctl.9s \
40 fmodsw.9s \
41 free_rtn.9s \
42 gld_mac_info.9s \
43 gld_stats.9s \
44 hook_nic_event.9s \
45 hook_pkt_event.9s \
46 hook_t.9s \
47 iocblk.9s \
48 iovec.9s \
49 kstat.9s \
50 kstat_intr.9s \
51 kstat_io.9s \
52 kstat_named.9s \
53 linkblk.9s \
54 modldrv.9s \
55 modlinkage.9s \
56 modlmisc.9s
```

new/usr/src/man/man9s/Makefile

2

```
54 modlstrmod.9s \
55 module_info.9s \
56 msgb.9s \
57 net_inject_t.9s \
58 net_instance_t.9s \
59 qband.9s \
60 qinit.9s \
61 queclass.9s \
62 queue.9s \
63 scsi_address.9s \
64 scsi_arq_status.9s \
65 scsi_asc_key_strings.9s \
66 scsi_device.9s \
67 scsi_extended_sense.9s \
68 scsi_hba_tran.9s \
69 scsi_inquiry.9s \
70 scsi_pkt.9s \
71 scsi_status.9s \
72 streamtab.9s \
73 stroptions.9s \
74 tuple.9s \
75 uio.9s \
76 usb_bulk_request.9s \
77 usb_callback_flags.9s \
78 usb_cfg_descr.9s \
79 usb_client_dev_data.9s \
80 usb_completion_reason.9s \
81 usb_ctrl_request.9s \
82 usb_dev_descr.9s \
83 usb_dev_qlf_descr.9s \
84 usb_ep_descr.9s \
85 usb_if_descr.9s \
86 usb_intr_request.9s \
87 usb_isoc_request.9s \
88 usb_other_speed_cfg_descr.9s \
89 usb_request_attributes.9s \
90 usb_string_descr.9s
```

```
92 MANLINKS= dblk.9s \
93 ddi_dma_lim.9s \
94 intro.9s \
95 mblk.9s
```

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

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

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

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

102 .KEEP_STATE:

104 include \$(SRC)/man/Makefile.man

106 install: \$(ROOTMANFILES) \$(ROOTMANLINKS)

```

*****
9917 Sat May 24 17:48:27 2014
new/usr/src/man/man9s/ddi_dmae_req.9s
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 \" te
2 .\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
3 .\" Copyright (c) 2004, Sun Microsystems, Inc., All Rights Reserved
4 .\" Copyright 2012 Garrett D'Amore <garrett@damore.org>. 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
7 .TH DDI_DMAE_REQ 9S \"May 24, 2014\"
7 .TH DDI_DMAE_REQ 9S \"Feb 06, 2012\"
8 .SH NAME
9 ddi_dmae_req \- DMA engine request structure
10 .SH SYNOPSIS
11 .LP
12 .nf
13 #include <sys/dma_engine.h>
14 .fi

16 .SH INTERFACE LEVEL
17 .sp
18 .LP
19 Solaris x86 DDI specific (Solaris x86 DDI).
20 .SH DESCRIPTION
21 .sp
22 .LP
23 A device driver uses the \fBddi_dmae_req\fR structure to describe the
24 parameters for a \fBDMA\fR channel. This structure contains all the information
25 necessary to set up the channel, except for the \fBDMA\fR memory address and
26 transfer count. The defaults, as specified below, support most standard
27 devices. Other modes might be desirable for some devices, or to increase
28 performance. The \fBDMA\fR engine request structure is passed to
29 \fBddi_dmae_prog\fR(9F).
30 .SH STRUCTURE MEMBERS
31 .sp
32 .LP
33 The \fBddi_dmae_req\fR structure contains several members, each of which
34 controls some aspect of DMA engine operation. The structure members associated
35 with supported DMA engine options are described here.
36 .sp
37 .in +2
38 .nf
39 uchar_tder_command; /* Read / Write *
40 /uchar_tder_bufprocess; /* Standard / Chain */
41 uchar_tder_path; /* 8 / 16 / 32 */
42 uchar_tder_cycles; /* Compat / Type A / Type B / Burst */
43 uchar_tder_trans; /* Single / Demand / Block */
44 ddi_dma_cookie_t*(*proc()); /* address of nextcookie routine */
45 void*procparms; /* parameter for nextcookie call */
46 .fi
47 .in -2

49 .sp
50 .ne 2
51 .na
52 \fB\bder_command\fR\bR
53 .ad
54 .RS 18n

```

```

55 Specifies what \fBDMA\fR operation is to be performed. The value
56 \fBDMAE_CMD_WRITE\fR signifies that data is to be transferred from memory to
57 the \fBI/O \fRdevice. The value \fBDMAE_CMD_READ\fR signifies that data is to
58 be transferred from the \fBI/O\fR device to memory. This field must be set by
59 the driver before calling \fBddi_dmae_prog()\fR.
60 .RE

62 .sp
63 .ne 2
64 .na
65 \fB\bder_bufprocess\fR\bR
66 .ad
67 .RS 18n
68 On some bus types, a driver can set \fBder_bufprocess\fR to the value
69 \fBDMAE_BUF_CHAIN\fR to specify that multiple \fBDMA\fR cookies will be given
70 to the \fBDMA\fR engine for a single \fBI/O\fR transfer. This action causes a
71 scatter/gather operation. In this mode of operation, the driver calls
72 \fBddi_dmae_prog()\fR to give the \fBDMA\fR engine the \fBDMA\fR engine request
73 structure and a pointer to the first cookie. The \fBproc\fR structure member
74 must be set to the address of a driver \fBnextcookie\fR routine. This routine
75 takes one argument, specified by the \fBprocparms\fR structure member, and
76 returns a pointer to a structure of type \fBddi_dma_cookie_t\fR that specifies
77 the next cookie for the \fBI/O \fRtransfer. When the \fBDMA\fR engine is ready
78 to receive an additional cookie, the bus nexus driver controlling that
79 \fBDMA\fR engine calls the routine specified by the \fBproc\fR structure member
80 to obtain the next cookie from the driver. The driver's \fBnextcookie\fR
81 routine must then return the address of the next cookie (in static storage) to
82 the bus nexus routine that called it. If there are no more segments in the
83 current \fBDMA\fR window, then \fB(*proc)()\fR must return the \fBNULL\fR
84 pointer.
85 .sp
86 A driver can specify the \fBDMAE_BUF_CHAIN\fR flag only if the particular bus
87 architecture supports the use of multiple \fBDMA\fR cookies in a single
88 \fBI/O\fR transfer. A bus \fBDMA\fR engine can support this feature either with
89 a fixed-length scatter/gather list, or by an interrupt chaining feature. A
90 driver must determine whether its parent bus nexus supports this feature by
91 examining the scatter/gather list size returned in the \fBdma_attr_sgllen\fR
92 member of the \fBDMA\fR attributes structure returned by the driver's call to
93 \fBddi_dmae_getattr()\fR. (See \fBddi_dma_attr\fR(9S).) If the size of the
94 examining the scatter/gather list size returned in the \fBdlim_sgllen\fR member
95 of the \fBDMA\fR limit structure returned by the driver's call to
96 \fBddi_dmae_getlim()\fR. (See \fBddi_dma_lim_x86\fR(9S).) If the size of the
97 scatter/gather list is 1, then no chaining is available. The driver must not
98 pass to \fBddi_dmae_prog()\fR, and the driver need not provide a
99 \fBnextcookie\fR routine.
100 .sp
101 If the size of the scatter/gather list is greater than 1, then \fBDMA\fR
102 chaining is available, and the driver has two options. Under the first option,
103 the driver chooses not to use the chaining feature. In this case (a) the driver
104 must \fBset\fR the size of the scatter/gather list to 1 before passing it to
105 the \fBDMAE_BUF_CHAIN\fR setup routine, and (b) the driver must not set the
106 \fBDMAE_BUF_CHAIN\fR flag.
107 .sp
108 Under the second option, the driver chooses to use the chaining feature, in
109 which case, (a) it should leave the size of the scatter/gather list alone, and
110 (b) it must set the \fBDMAE_BUF_CHAIN\fR flag in the \fBddi_dmae_req\fR
111 structure. Before calling \fBddi_dmae_prog()\fR, the driver must \fBprefetch\fR
112 cookies until either (1) the end of the DMA window is
113 reached, or (2) the size of the
114 scatter/gather list is reached, whichever occurs first. These cookies must be
115 saved by the driver until they are requested by the nexus driver calling the
116 driver's \fBnextcookie\fR routine. The driver's \fBnextcookie\fR routine must
117 return the prefetched cookies in order, one cookie for each call to the
118 \fBnextcookie\fR routine, until the list of prefetched cookies is exhausted.
119 After the end of the list of cookies is reached, the \fBnextcookie\fR routine

```

```

118 must return the \fBNULL\fr pointer.
119 .sp
120 The size of the scatter/gather list determines how many discontinuous segments
121 of physical memory can participate in a single \fBDMA\fr transfer. \fBISA\fr
122 bus \fBDMA\fr engines have no scatter/gather capability, so their
123 scatter/gather list sizes are 1. Other finite scatter/gather list sizes would
124 also be possible. For performance reasons, drivers should use the chaining
125 capability if it is available on their parent bus.
126 .sp
127 As described above, a driver making use of \fBDMA\fr chaining must prefetch
128 \fBDMA\fr cookies before calling \fBddi_dmae_prog()\fr. The reasons for this
129 are:
130 .RS +4
131 .TP
132 .ie t \(\bu
133 .el o
134 First, the driver must have some way to know the total \fBI/O\fr count with
135 which to program the \fBI/O\fr device. This \fBI/O\fr count must match the
136 total size of all the \fBDMA\fr segments that will be chained together into one
137 \fBDMA\fr operation. Depending on the size of the scatter/gather list and the
138 memory position and alignment of the \fBDMA\fr object, all or just part of the
139 current \fBDMA\fr window might be able to participate in a single \fBI/O\fr
140 operation. The driver must compute the \fBI/O\fr count by adding up the sizes
141 of the prefetched \fBDMA\fr cookies. The number of cookies whose sizes are to
142 be summed is the lesser of (a) the size of the scatter/gather list, or (b) the
143 number of segments remaining in the window.
144 .RE
145 .RS +4
146 .TP
147 .ie t \(\bu
148 .el o
149 Second, on some bus architectures, the driver's \fBnextcookie\fr routine can be
150 called from a high-level interrupt routine. If the cookies were not prefetched,
151 the \fBnextcookie\fr routine would have to call \fBDMA\fr functions
152 from a high-level interrupt routine, which is not
153 recommended.
154 .RE
155 When breaking a \fBDMA\fr window into segments, the system arranges for the end
156 of every segment whose number is an integral multiple of the scatter/gather
157 list size to fall on a device-granularity boundary, as specified in the
158 \fBdma_attr_granular\fr field in the \fBddi_dma_attr\fr(9S) structure.
159 \fBdlim_granular\fr field in the \fBddi_dma_lim_x86\fr(9S) structure.
160 .sp
161 If the scatter/gather list size is 1 (either because no chaining is available
162 or because the driver does not want to use the chaining feature), then the
163 total \fBI/O\fr count for a single \fBDMA\fr operation is the size of \fBDMA\fr
164 segment denoted by the single \fBDMA\fr cookie that is passed in the call to
165 \fBddi_dmae_prog()\fr. In this case, the system arranges for each \fBDMA\fr
166 segment to be a multiple of the device-granularity size.
167 .RE
168 .sp
169 .ne 2
170 .na
171 \fB\fbder_path\fr\fr
172 .ad
173 .RS 18n
174 Specifies the \fBDMA\fr transfer size. The default of zero
175 (\fBDMAE_PATH_DEF\fr) specifies \fBISA\fr compatibility mode. In that mode,
176 channels 0, 1, 2, and 3 are programmed in 8-bit mode (\fBDMAE_PATH_8\fr), and
177 channels 5, 6, and 7 are programmed in 16-bit, count-by-word mode
178 (\fBDMAE_PATH_16\fr).
179 .RE
181 .sp
182 .ne 2

```

```

183 .na
184 \fB\fbder_cycles\fr\fr
185 .ad
186 .RS 18n
187 Specifies the timing mode to be used during \fBDMA\fr data transfers. The
188 default of zero (\fBDMAE_CYCLES_1\fr) specifies \fBISA\fr compatible timing.
189 Drivers using this mode must also specify \fBDMAE_TRANS_SINGL\fr in the
190 \fBder_trans\fr structure member.
191 .RE
193 .sp
194 .ne 2
195 .na
196 \fB\fbder_trans\fr\fr
197 .ad
198 .RS 18n
199 Specifies the bus transfer mode that the \fBDMA\fr engine should expect from
200 the device. The default value of zero (\fBDMAE_TRANS_SINGL\fr) specifies that
201 the device performs one transfer for each bus arbitration cycle. Devices that
202 use \fBISA\fr compatible timing (specified by a value of zero, which is the
203 default, in the \fBder_cycles\fr structure member) should use the
204 \fBDMAE_TRANS_SINGL\fr mode.
205 .RE
207 .SH ATTRIBUTES
208 .sp
209 .LP
210 See \fBattributes\fr(5) for descriptions of the following attributes:
211 .sp
213 .sp
214 .TS
215 box;
216 c | c
217 l | l
218 ATTRIBUTE TYPE ATTRIBUTE VALUE
219 _
220 Architecture x86
221 .TE
223 .SH SEE ALSO
224 .sp
225 .LP
226 \fBisa\fr(4), \fBattributes\fr(5),
227 \fBddi_dmae\fr(9F), \fBddi_dma_attr\fr(9S)
228 \fBddi_dmae\fr(9F), \fBddi_dma_lim_x86\fr(9S), \fBddi_dma_req\fr(9S)

```

68343 Sat May 24 17:48:27 2014

new/usr/src/pkg/manifests/system-kernel.man9f.inc

4888 Undocument dma_req(9s)

4884 EOF scsi_hba_attach

4886 EOF ddi_dmae_getlim

4887 EOF ddi_iomin

4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)

4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free

1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet
9 # at http://www.illumos.org/license/CDDL.
10 #
11 #
12 #
13 # Copyright 2011, Richard Lowe
14 # Copyright 2012 Garrett D'Amore <garrett@damore.org>. All rights reserved.
15 # Copyright 2012 Nexenta Systems, Inc. All rights reserved.
16 # Copyright 2014 Garrett D'Amore <garrett@damore.org>

18 file path=usr/share/man/man9f/ASSERT.9f
19 file path=usr/share/man/man9f/Intro.9f
20 file path=usr/share/man/man9f/OTHERQ.9f
21 file path=usr/share/man/man9f/RD.9f
22 file path=usr/share/man/man9f/SAMESTR.9f
23 file path=usr/share/man/man9f/STRUCT_DECL.9f
24 file path=usr/share/man/man9f/WR.9f
25 file path=usr/share/man/man9f/adjmsg.9f
26 file path=usr/share/man/man9f/allpcb.9f
27 file path=usr/share/man/man9f/atomic_add.9f
28 file path=usr/share/man/man9f/atomic_and.9f
29 file path=usr/share/man/man9f/atomic_bits.9f
30 file path=usr/share/man/man9f/atomic_cas.9f
31 file path=usr/share/man/man9f/atomic_dec.9f
32 file path=usr/share/man/man9f/atomic_inc.9f
33 file path=usr/share/man/man9f/atomic_ops.9f
34 file path=usr/share/man/man9f/atomic_or.9f
35 file path=usr/share/man/man9f/atomic_swap.9f
36 file path=usr/share/man/man9f/backq.9f
37 file path=usr/share/man/man9f/bcanput.9f
38 file path=usr/share/man/man9f/bcmp.9f
39 file path=usr/share/man/man9f/bcopy.9f
40 file path=usr/share/man/man9f/bioclone.9f
41 file path=usr/share/man/man9f/biodone.9f
42 file path=usr/share/man/man9f/bioerror.9f
43 file path=usr/share/man/man9f/biofini.9f
44 file path=usr/share/man/man9f/bioinit.9f
45 file path=usr/share/man/man9f/biomodified.9f
46 file path=usr/share/man/man9f/bioreset.9f
47 file path=usr/share/man/man9f/biosize.9f
48 file path=usr/share/man/man9f/biowait.9f
49 file path=usr/share/man/man9f/bp_copyin.9f
50 file path=usr/share/man/man9f/bp_copyout.9f
51 file path=usr/share/man/man9f/bp_mapin.9f
52 file path=usr/share/man/man9f/bp_mapout.9f
53 file path=usr/share/man/man9f/btop.9f
54 file path=usr/share/man/man9f/btopr.9f
55 file path=usr/share/man/man9f/bufcall.9f

56 file path=usr/share/man/man9f/bzero.9f
57 file path=usr/share/man/man9f/canput.9f
58 file path=usr/share/man/man9f/clrbuf.9f
59 file path=usr/share/man/man9f/cmn_err.9f
60 file path=usr/share/man/man9f/condvar.9f
61 file path=usr/share/man/man9f/copyb.9f
62 file path=usr/share/man/man9f/copyin.9f
63 file path=usr/share/man/man9f/copymsg.9f
64 file path=usr/share/man/man9f/copyout.9f
65 file path=usr/share/man/man9f/csx_AccessConfigurationRegister.9f
66 file path=usr/share/man/man9f/csx_CS_DDI_Info.9f
67 file path=usr/share/man/man9f/csx_ConvertSize.9f
68 file path=usr/share/man/man9f/csx_ConvertSpeed.9f
69 file path=usr/share/man/man9f/csx_DeregisterClient.9f
70 file path=usr/share/man/man9f/csx_DupHandle.9f
71 file path=usr/share/man/man9f/csx_Error2Text.9f
72 file path=usr/share/man/man9f/csx_Event2Text.9f
73 file path=usr/share/man/man9f/csx_FreeHandle.9f
74 file path=usr/share/man/man9f/csx_Get8.9f
75 file path=usr/share/man/man9f/csx_GetFirstClient.9f
76 file path=usr/share/man/man9f/csx_GetFirstTuple.9f
77 file path=usr/share/man/man9f/csx_GetHandleOffset.9f
78 file path=usr/share/man/man9f/csx_GetMappedAddr.9f
79 file path=usr/share/man/man9f/csx_GetStatus.9f
80 file path=usr/share/man/man9f/csx_GetTupleData.9f
81 file path=usr/share/man/man9f/csx_MakeDeviceNode.9f
82 file path=usr/share/man/man9f/csx_MapLogSocket.9f
83 file path=usr/share/man/man9f/csx_MapMemPage.9f
84 file path=usr/share/man/man9f/csx_ModifyConfiguration.9f
85 file path=usr/share/man/man9f/csx_ModifyWindow.9f
86 file path=usr/share/man/man9f/csx_ParseTuple.9f
87 file path=usr/share/man/man9f/csx_Parse_CISTPL_BATTERY.9f
88 file path=usr/share/man/man9f/csx_Parse_CISTPL_BYTEORDER.9f
89 file path=usr/share/man/man9f/csx_Parse_CISTPL_CFTABLE_ENTRY.9f
90 file path=usr/share/man/man9f/csx_Parse_CISTPL_CONFIG.9f
91 file path=usr/share/man/man9f/csx_Parse_CISTPL_DATE.9f
92 file path=usr/share/man/man9f/csx_Parse_CISTPL_DEVICE.9f
93 file path=usr/share/man/man9f/csx_Parse_CISTPL_DEVICEGEO.9f
94 file path=usr/share/man/man9f/csx_Parse_CISTPL_DEVICEGEO_A.9f
95 file path=usr/share/man/man9f/csx_Parse_CISTPL_FORMAT.9f
96 file path=usr/share/man/man9f/csx_Parse_CISTPL_FUNCCE.9f
97 file path=usr/share/man/man9f/csx_Parse_CISTPL_FUNCID.9f
98 file path=usr/share/man/man9f/csx_Parse_CISTPL_GEOMETRY.9f
99 file path=usr/share/man/man9f/csx_Parse_CISTPL_JEDEC_C.9f
100 file path=usr/share/man/man9f/csx_Parse_CISTPL_LINKTARGET.9f
101 file path=usr/share/man/man9f/csx_Parse_CISTPL_LONGLINK_A.9f
102 file path=usr/share/man/man9f/csx_Parse_CISTPL_LONGLINK_MFC.9f
103 file path=usr/share/man/man9f/csx_Parse_CISTPL_MNFID.9f
104 file path=usr/share/man/man9f/csx_Parse_CISTPL_ORG.9f
105 file path=usr/share/man/man9f/csx_Parse_CISTPL_SPCPL.9f
106 file path=usr/share/man/man9f/csx_Parse_CISTPL_SWLL.9f
107 file path=usr/share/man/man9f/csx_Parse_CISTPL_VERS_1.9f
108 file path=usr/share/man/man9f/csx_Parse_CISTPL_VERS_2.9f
109 file path=usr/share/man/man9f/csx_Put8.9f
110 file path=usr/share/man/man9f/csx_RegisterClient.9f
111 file path=usr/share/man/man9f/csx_ReleaseConfiguration.9f
112 file path=usr/share/man/man9f/csx_RepGet8.9f
113 file path=usr/share/man/man9f/csx_RepPut8.9f
114 file path=usr/share/man/man9f/csx_RequestConfiguration.9f
115 file path=usr/share/man/man9f/csx_RequestIO.9f
116 file path=usr/share/man/man9f/csx_RequestIRQ.9f
117 file path=usr/share/man/man9f/csx_RequestSocketMask.9f
118 file path=usr/share/man/man9f/csx_RequestWindow.9f
119 file path=usr/share/man/man9f/csx_ResetFunction.9f
120 file path=usr/share/man/man9f/csx_SetEventMask.9f
121 file path=usr/share/man/man9f/csx_SetHandleOffset.9f

```

122 file path=usr/share/man/man9f/csx_ValidateCIS.9f
123 file path=usr/share/man/man9f/datams9.9f
124 file path=usr/share/man/man9f/ddi_add_event_handler.9f
125 file path=usr/share/man/man9f/ddi_add_intr.9f
126 file path=usr/share/man/man9f/ddi_add_softintr.9f
127 file path=usr/share/man/man9f/ddi_binding_name.9f
128 file path=usr/share/man/man9f/ddi_btop.9f
129 file path=usr/share/man/man9f/ddi_can_receive_sig.9f
130 file path=usr/share/man/man9f/ddi_cb_register.9f
131 file path=usr/share/man/man9f/ddi_check_acc_handle.9f
132 file path=usr/share/man/man9f/ddi_copyin.9f
133 file path=usr/share/man/man9f/ddi_copyout.9f
134 file path=usr/share/man/man9f/ddi_create_minor_node.9f
135 file path=usr/share/man/man9f/ddi_cred.9f
136 file path=usr/share/man/man9f/ddi_dev_is_needed.9f
137 file path=usr/share/man/man9f/ddi_dev_is_sid.9f
138 file path=usr/share/man/man9f/ddi_dev_nintrs.9f
139 file path=usr/share/man/man9f/ddi_dev_nregs.9f
140 file path=usr/share/man/man9f/ddi_dev_regsize.9f
141 file path=usr/share/man/man9f/ddi_dev_report_fault.9f
142 file path=usr/share/man/man9f/ddi_device_copy.9f
143 file path=usr/share/man/man9f/ddi_device_zero.9f
144 file path=usr/share/man/man9f/ddi_devid_compare.9f
145 file path=usr/share/man/man9f/ddi_dma_addr_bind_handle.9f
146 file path=usr/share/man/man9f/ddi_dma_alloc_handle.9f
147 file path=usr/share/man/man9f/ddi_dma_buf_bind_handle.9f
148 file path=usr/share/man/man9f/ddi_dma_burstsizes.9f
149 file path=usr/share/man/man9f/ddi_dma_free_handle.9f
150 file path=usr/share/man/man9f/ddi_dma_getwin.9f
151 file path=usr/share/man/man9f/ddi_dma_mem_alloc.9f
152 file path=usr/share/man/man9f/ddi_dma_mem_free.9f
153 file path=usr/share/man/man9f/ddi_dma_nextcookie.9f
154 file path=usr/share/man/man9f/ddi_dma_numwin.9f
155 file path=usr/share/man/man9f/ddi_dma_set_sbus64.9f
156 file path=usr/share/man/man9f/ddi_dma_sync.9f
157 file path=usr/share/man/man9f/ddi_dma_unbind_handle.9f
158 file path=usr/share/man/man9f/ddi_dmae.9f
159 file path=usr/share/man/man9f/ddi_driver_major.9f
160 file path=usr/share/man/man9f/ddi_driver_name.9f
161 file path=usr/share/man/man9f/ddi_enter_critical.9f
162 file path=usr/share/man/man9f/ddi_ffs.9f
163 file path=usr/share/man/man9f/ddi_fm_acc_err_clear.9f
164 file path=usr/share/man/man9f/ddi_fm_acc_err_get.9f
165 file path=usr/share/man/man9f/ddi_fm_ereport_post.9f
166 file path=usr/share/man/man9f/ddi_fm_handler_register.9f
167 file path=usr/share/man/man9f/ddi_fm_init.9f
168 file path=usr/share/man/man9f/ddi_fm_service_impact.9f
169 file path=usr/share/man/man9f/ddi_get8.9f
170 file path=usr/share/man/man9f/ddi_get_cred.9f
171 file path=usr/share/man/man9f/ddi_get_devstate.9f
172 file path=usr/share/man/man9f/ddi_get_driver_private.9f
173 file path=usr/share/man/man9f/ddi_get_eventcookie.9f
174 file path=usr/share/man/man9f/ddi_get_instance.9f
175 file path=usr/share/man/man9f/ddi_get_kt_did.9f
176 file path=usr/share/man/man9f/ddi_get_lbolt.9f
177 file path=usr/share/man/man9f/ddi_get_parent.9f
178 file path=usr/share/man/man9f/ddi_get_pid.9f
179 file path=usr/share/man/man9f/ddi_get_time.9f
180 file path=usr/share/man/man9f/ddi_getiminor.9f
181 file path=usr/share/man/man9f/ddi_in_panic.9f
182 file path=usr/share/man/man9f/ddi_intr_add_handler.9f
183 file path=usr/share/man/man9f/ddi_intr_add_softint.9f
184 file path=usr/share/man/man9f/ddi_intr_alloc.9f
185 file path=usr/share/man/man9f/ddi_intr_dup_handler.9f
186 file path=usr/share/man/man9f/ddi_intr_enable.9f
187 file path=usr/share/man/man9f/ddi_intr_get_cap.9f

```

```

188 file path=usr/share/man/man9f/ddi_intr_get_hilevel_pri.9f
189 file path=usr/share/man/man9f/ddi_intr_get_nintrs.9f
190 file path=usr/share/man/man9f/ddi_intr_get_pending.9f
191 file path=usr/share/man/man9f/ddi_intr_get_pri.9f
192 file path=usr/share/man/man9f/ddi_intr_get_supported_types.9f
193 file path=usr/share/man/man9f/ddi_intr_hilevel.9f
194 file path=usr/share/man/man9f/ddi_intr_set_mask.9f
195 file path=usr/share/man/man9f/ddi_intr_set_nreq.9f
196 file path=usr/share/man/man9f/ddi_io_get8.9f
197 file path=usr/share/man/man9f/ddi_io_put8.9f
198 file path=usr/share/man/man9f/ddi_io_rep_get8.9f
199 file path=usr/share/man/man9f/ddi_io_rep_put8.9f
200 file path=usr/share/man/man9f/ddi_iomin.9f
201 file path=usr/share/man/man9f/ddi_log_sysevent.9f
202 file path=usr/share/man/man9f/ddi_map_regs.9f
203 file path=usr/share/man/man9f/ddi_mem_get8.9f
204 file path=usr/share/man/man9f/ddi_mem_put8.9f
205 file path=usr/share/man/man9f/ddi_mem_rep_get8.9f
206 file path=usr/share/man/man9f/ddi_mmap_get_model.9f
207 file path=usr/share/man/man9f/ddi_model_convert_from.9f
208 file path=usr/share/man/man9f/ddi_modopen.9f
209 file path=usr/share/man/man9f/ddi_no_info.9f
210 file path=usr/share/man/man9f/ddi_node_name.9f
211 file path=usr/share/man/man9f/ddi_peek.9f
212 file path=usr/share/man/man9f/ddi_periodic_add.9f
213 file path=usr/share/man/man9f/ddi_periodic_delete.9f
214 file path=usr/share/man/man9f/ddi_poke.9f
215 file path=usr/share/man/man9f/ddi_prop_create.9f
216 file path=usr/share/man/man9f/ddi_prop_exists.9f
217 file path=usr/share/man/man9f/ddi_prop_get_int.9f
218 file path=usr/share/man/man9f/ddi_prop_lookup.9f
219 file path=usr/share/man/man9f/ddi_prop_op.9f
220 file path=usr/share/man/man9f/ddi_prop_update.9f
221 file path=usr/share/man/man9f/ddi_put8.9f
222 file path=usr/share/man/man9f/ddi_regs_map_free.9f
223 file path=usr/share/man/man9f/ddi_regs_map_setup.9f
224 file path=usr/share/man/man9f/ddi_remove_event_handler.9f
225 file path=usr/share/man/man9f/ddi_remove_minor_node.9f
226 file path=usr/share/man/man9f/ddi_removing_power.9f
227 file path=usr/share/man/man9f/ddi_rep_get8.9f
228 file path=usr/share/man/man9f/ddi_rep_put8.9f
229 file path=usr/share/man/man9f/ddi_report_dev.9f
230 file path=usr/share/man/man9f/ddi_root_node.9f
231 file path=usr/share/man/man9f/ddi_segmap.9f
232 file path=usr/share/man/man9f/ddi_slaveonly.9f
233 file path=usr/share/man/man9f/ddi_soft_state.9f
234 file path=usr/share/man/man9f/ddi_strtol.9f
235 file path=usr/share/man/man9f/ddi_strtol.9f
236 file path=usr/share/man/man9f/ddi_strtoul.9f
237 file path=usr/share/man/man9f/ddi_umem_alloc.9f
238 file path=usr/share/man/man9f/ddi_umem_iosetup.9f
239 file path=usr/share/man/man9f/ddi_umem_lock.9f
240 file path=usr/share/man/man9f/delay.9f
241 file path=usr/share/man/man9f/devmap_default_access.9f
242 file path=usr/share/man/man9f/devmap_devmem_setup.9f
243 file path=usr/share/man/man9f/devmap_do_ctxmgt.9f
244 file path=usr/share/man/man9f/devmap_set_ctx_timeout.9f
245 file path=usr/share/man/man9f/devmap_setup.9f
246 file path=usr/share/man/man9f/devmap_unload.9f
247 file path=usr/share/man/man9f/disksort.9f
248 file path=usr/share/man/man9f/dlbindack.9f
249 file path=usr/share/man/man9f/drv_getparm.9f
250 file path=usr/share/man/man9f/drv_hztousec.9f
251 file path=usr/share/man/man9f/drv_priv.9f
252 file path=usr/share/man/man9f/drv_usctohz.9f

```

new/usr/src/pkg/manifests/system-kernel.man9f.inc

5

```
253 file path=usr/share/man/man9f/drv_useccwait.9f
254 file path=usr/share/man/man9f/dupb.9f
255 file path=usr/share/man/man9f/dupmsg.9f
256 file path=usr/share/man/man9f/enablelok.9f
257 file path=usr/share/man/man9f/esballoc.9f
258 file path=usr/share/man/man9f/esbcall.9f
259 file path=usr/share/man/man9f/flushband.9f
260 file path=usr/share/man/man9f/flushq.9f
261 file path=usr/share/man/man9f/freeb.9f
262 file path=usr/share/man/man9f/freemsg.9f
263 file path=usr/share/man/man9f/freerbuf.9f
264 file path=usr/share/man/man9f/freezeestr.9f
265 file path=usr/share/man/man9f/get_pktiopb.9f
266 file path=usr/share/man/man9f/geterror.9f
267 file path=usr/share/man/man9f/gethrtime.9f
268 file path=usr/share/man/man9f/getmajor.9f
269 file path=usr/share/man/man9f/getminor.9f
270 file path=usr/share/man/man9f/getq.9f
271 file path=usr/share/man/man9f/getrbuf.9f
272 file path=usr/share/man/man9f/gld.9f
273 file path=usr/share/man/man9f/hook_alloc.9f
274 file path=usr/share/man/man9f/hook_free.9f
275 file path=usr/share/man/man9f/id32_alloc.9f
276 file path=usr/share/man/man9f/inb.9f
277 file path=usr/share/man/man9f/insq.9f
278 file path=usr/share/man/man9f/kiconv.9f
279 file path=usr/share/man/man9f/kiconv_close.9f
280 file path=usr/share/man/man9f/kiconv_open.9f
281 file path=usr/share/man/man9f/kiconvstr.9f
282 file path=usr/share/man/man9f/kmem_alloc.9f
283 file path=usr/share/man/man9f/kmem_cache_create.9f
284 file path=usr/share/man/man9f/kstat_create.9f
285 file path=usr/share/man/man9f/kstat_delete.9f
286 file path=usr/share/man/man9f/kstat_install.9f
287 file path=usr/share/man/man9f/kstat_named_init.9f
288 file path=usr/share/man/man9f/kstat_queue.9f
289 file path=usr/share/man/man9f/ldi_add_event_handler.9f
290 file path=usr/share/man/man9f/ldi_aread.9f
291 file path=usr/share/man/man9f/ldi_devmap.9f
292 file path=usr/share/man/man9f/ldi_dump.9f
293 file path=usr/share/man/man9f/ldi_ev_finalize.9f
294 file path=usr/share/man/man9f/ldi_ev_get_cookie.9f
295 file path=usr/share/man/man9f/ldi_ev_get_type.9f
296 file path=usr/share/man/man9f/ldi_ev_notify.9f
297 file path=usr/share/man/man9f/ldi_ev_register_callbacks.9f
298 file path=usr/share/man/man9f/ldi_ev_remove_callbacks.9f
299 file path=usr/share/man/man9f/ldi_get_dev.9f
300 file path=usr/share/man/man9f/ldi_get_eventcookie.9f
301 file path=usr/share/man/man9f/ldi_get_size.9f
302 file path=usr/share/man/man9f/ldi_ident_from_dev.9f
303 file path=usr/share/man/man9f/ldi_ioctl.9f
304 file path=usr/share/man/man9f/ldi_open_by_dev.9f
305 file path=usr/share/man/man9f/ldi_poll.9f
306 file path=usr/share/man/man9f/ldi_prop_exists.9f
307 file path=usr/share/man/man9f/ldi_prop_get_int.9f
308 file path=usr/share/man/man9f/ldi_prop_lookup_int_array.9f
309 file path=usr/share/man/man9f/ldi_putmsg.9f
310 file path=usr/share/man/man9f/ldi_read.9f
311 file path=usr/share/man/man9f/ldi_remove_event_handler.9f
312 file path=usr/share/man/man9f/ldi_strategy.9f
313 file path=usr/share/man/man9f/linkb.9f
314 file path=usr/share/man/man9f/list_create.9f
315 file path=usr/share/man/man9f/makecom.9f
316 file path=usr/share/man/man9f/makedevice.9f
317 file path=usr/share/man/man9f/max.9f
318 file path=usr/share/man/man9f/mcopyin.9f
```

new/usr/src/pkg/manifests/system-kernel.man9f.inc

6

```
319 file path=usr/share/man/man9f/mcopymsg.9f
320 file path=usr/share/man/man9f/mcopyout.9f
321 file path=usr/share/man/man9f/membar_ops.9f
322 file path=usr/share/man/man9f/memchr.9f
323 file path=usr/share/man/man9f/merror.9f
324 file path=usr/share/man/man9f/mexchange.9f
325 file path=usr/share/man/man9f/min.9f
326 file path=usr/share/man/man9f/mioc2ack.9f
327 file path=usr/share/man/man9f/miocack.9f
328 file path=usr/share/man/man9f/miocnak.9f
329 file path=usr/share/man/man9f/miocpullup.9f
330 file path=usr/share/man/man9f/mkiocb.9f
331 file path=usr/share/man/man9f/mod_install.9f
332 file path=usr/share/man/man9f/msgdsize.9f
333 file path=usr/share/man/man9f/msgpullup.9f
334 file path=usr/share/man/man9f/msgsize.9f
335 file path=usr/share/man/man9f/mt-streams.9f
336 file path=usr/share/man/man9f/mutex.9f
337 file path=usr/share/man/man9f/net_event_notify_register.9f
338 file path=usr/share/man/man9f/net_getifname.9f
339 file path=usr/share/man/man9f/net_getlifaddr.9f
340 file path=usr/share/man/man9f/net_getmtu.9f
341 file path=usr/share/man/man9f/net_getnetid.9f
342 file path=usr/share/man/man9f/net_getpmtuenabled.9f
343 file path=usr/share/man/man9f/net_hook_register.9f
344 file path=usr/share/man/man9f/net_hook_unregister.9f
345 file path=usr/share/man/man9f/net_inject.9f
346 file path=usr/share/man/man9f/net_inject_alloc.9f
347 file path=usr/share/man/man9f/net_inject_free.9f
348 file path=usr/share/man/man9f/net_instance_alloc.9f
349 file path=usr/share/man/man9f/net_instance_free.9f
350 file path=usr/share/man/man9f/net_instance_notify_register.9f
351 file path=usr/share/man/man9f/net_instance_register.9f
352 file path=usr/share/man/man9f/net_instance_unregister.9f
353 file path=usr/share/man/man9f/net_ispartialchecksum.9f
354 file path=usr/share/man/man9f/net_isvalidchecksum.9f
355 file path=usr/share/man/man9f/net_kstat_create.9f
356 file path=usr/share/man/man9f/net_kstat_delete.9f
357 file path=usr/share/man/man9f/net_lifgetnext.9f
358 file path=usr/share/man/man9f/net_netidtozonid.9f
359 file path=usr/share/man/man9f/net_phygetnext.9f
360 file path=usr/share/man/man9f/net_phylookup.9f
361 file path=usr/share/man/man9f/net_protocol_lookup.9f
362 file path=usr/share/man/man9f/net_protocol_notify_register.9f
363 file path=usr/share/man/man9f/net_protocol_release.9f
364 file path=usr/share/man/man9f/net_protocol_walk.9f
365 file path=usr/share/man/man9f/net_routeto.9f
366 file path=usr/share/man/man9f/net_zoneidtonetid.9f
367 file path=usr/share/man/man9f/netinfo.9f
368 file path=usr/share/man/man9f/nochpoll.9f
369 file path=usr/share/man/man9f/nodev.9f
370 file path=usr/share/man/man9f/noenable.9f
371 file path=usr/share/man/man9f/nulldev.9f
372 file path=usr/share/man/man9f/nvlist_add_boolean.9f
373 file path=usr/share/man/man9f/nvlist_alloc.9f
374 file path=usr/share/man/man9f/nvlist_lookup_boolean.9f
375 file path=usr/share/man/man9f/nvlist_lookup_nvpair.9f
376 file path=usr/share/man/man9f/nvlist_next_nvpair.9f
377 file path=usr/share/man/man9f/nvlist_remove.9f
378 file path=usr/share/man/man9f/nvpair_value_byte.9f
379 file path=usr/share/man/man9f/outb.9f
380 file path=usr/share/man/man9f/pci_config_get8.9f
381 file path=usr/share/man/man9f/pci_config_setup.9f
382 file path=usr/share/man/man9f/pci_ereport_setup.9f
383 file path=usr/share/man/man9f/pci_report_pmcap.9f
384 file path=usr/share/man/man9f/pci_save_config_regs.9f
```

```

385 file path=usr/share/man/man9f/physio.9f
386 file path=usr/share/man/man9f/pm_busy_component.9f
387 file path=usr/share/man/man9f/pm_power_has_changed.9f
388 file path=usr/share/man/man9f/pm_raise_power.9f
389 file path=usr/share/man/man9f/pm_trans_check.9f
390 file path=usr/share/man/man9f/pollwakeupp.9f
391 file path=usr/share/man/man9f/priv_getbyname.9f
392 file path=usr/share/man/man9f/priv_policy.9f
393 file path=usr/share/man/man9f/proc_signal.9f
394 file path=usr/share/man/man9f/ptob.9f
395 file path=usr/share/man/man9f/pullupmsg.9f
396 file path=usr/share/man/man9f/put.9f
397 file path=usr/share/man/man9f/putbq.9f
398 file path=usr/share/man/man9f/putctl.9f
399 file path=usr/share/man/man9f/putctl1.9f
400 file path=usr/share/man/man9f/putnext.9f
401 file path=usr/share/man/man9f/putnextctl.9f
402 file path=usr/share/man/man9f/putnextctl1.9f
403 file path=usr/share/man/man9f/putq.9f
404 file path=usr/share/man/man9f/qassociate.9f
405 file path=usr/share/man/man9f/qbufcall.9f
406 file path=usr/share/man/man9f/qenable.9f
407 file path=usr/share/man/man9f/qprocson.9f
408 file path=usr/share/man/man9f/qreply.9f
409 file path=usr/share/man/man9f/qsize.9f
410 file path=usr/share/man/man9f/qtimeout.9f
411 file path=usr/share/man/man9f/qunbufcall.9f
412 file path=usr/share/man/man9f/quntimeout.9f
413 file path=usr/share/man/man9f/qwait.9f
414 file path=usr/share/man/man9f/qwriter.9f
415 file path=usr/share/man/man9f/rmalloc.9f
416 file path=usr/share/man/man9f/rmalloc_wait.9f
417 file path=usr/share/man/man9f/rmallocmap.9f
418 file path=usr/share/man/man9f/rmfree.9f
419 file path=usr/share/man/man9f/rmvb.9f
420 file path=usr/share/man/man9f/rmvb.9f
421 file path=usr/share/man/man9f/rwlock.9f
422 file path=usr/share/man/man9f/scsi_abort.9f
423 file path=usr/share/man/man9f/scsi_alloc_consistent_buf.9f
424 file path=usr/share/man/man9f/scsi_cname.9f
425 file path=usr/share/man/man9f/scsi_destroy_pkt.9f
426 file path=usr/share/man/man9f/scsi_dmaget.9f
427 file path=usr/share/man/man9f/scsi_errmsg.9f
428 file path=usr/share/man/man9f/scsi_ext_sense_fields.9f
429 file path=usr/share/man/man9f/scsi_find_sense_descr.9f
430 file path=usr/share/man/man9f/scsi_free_consistent_buf.9f
431 file path=usr/share/man/man9f/scsi_get_device_type_scsi_options.9f
432 file path=usr/share/man/man9f/scsi_get_device_type_string.9f
433 file path=usr/share/man/man9f/scsi_hba_attach_setup.9f
434 file path=usr/share/man/man9f/scsi_hba_init.9f
435 file path=usr/share/man/man9f/scsi_hba_lookup_capstr.9f
436 file path=usr/share/man/man9f/scsi_hba_pkt_alloc.9f
437 file path=usr/share/man/man9f/scsi_hba_pkt_comp.9f
438 file path=usr/share/man/man9f/scsi_hba_probe.9f
439 file path=usr/share/man/man9f/scsi_hba_tran_alloc.9f
440 file path=usr/share/man/man9f/scsi_ifgetcap.9f
441 file path=usr/share/man/man9f/scsi_init_pkt.9f
442 file path=usr/share/man/man9f/scsi_log.9f
443 file path=usr/share/man/man9f/scsi_pktalloc.9f
444 file path=usr/share/man/man9f/scsi_poll.9f
445 file path=usr/share/man/man9f/scsi_probe.9f
446 file path=usr/share/man/man9f/scsi_reset.9f
447 file path=usr/share/man/man9f/scsi_reset_notify.9f
448 file path=usr/share/man/man9f/scsi_sense_key.9f
449 file path=usr/share/man/man9f/scsi_setup_cdb.9f
450 file path=usr/share/man/man9f/scsi_slave.9f

```

```

451 file path=usr/share/man/man9f/scsi_sync_pkt.9f
452 file path=usr/share/man/man9f/scsi_transport.9f
453 file path=usr/share/man/man9f/scsi_unprobe.9f
454 file path=usr/share/man/man9f/scsi_validate_sense.9f
455 file path=usr/share/man/man9f/scsi_vu_errmsg.9f
456 file path=usr/share/man/man9f/semaphore.9f
457 file path=usr/share/man/man9f/stoi.9f
458 file path=usr/share/man/man9f/string.9f
459 file path=usr/share/man/man9f/strlog.9f
460 file path=usr/share/man/man9f/strqget.9f
461 file path=usr/share/man/man9f/strqset.9f
462 file path=usr/share/man/man9f/swab.9f
463 file path=usr/share/man/man9f/taskq.9f
464 file path=usr/share/man/man9f/testb.9f
465 file path=usr/share/man/man9f/timeout.9f
466 file path=usr/share/man/man9f/u8_strcmp.9f
467 file path=usr/share/man/man9f/u8_textprep_str.9f
468 file path=usr/share/man/man9f/u8_validate.9f
469 file path=usr/share/man/man9f/uconv_ul6tou32.9f
470 file path=usr/share/man/man9f/uiomove.9f
471 file path=usr/share/man/man9f/unbufcall.9f
472 file path=usr/share/man/man9f/unlinkb.9f
473 file path=usr/share/man/man9f/untimeout.9f
474 file path=usr/share/man/man9f/ureadc.9f
475 file path=usr/share/man/man9f/urwritec.9f
476 file path=usr/share/man/man9f/va_arg.9f
477 file path=usr/share/man/man9f/vsprintf.9f
478 link path=usr/share/man/man9f/SIZEOF_PTR.9f target=STRUCT_DECL.9f
479 link path=usr/share/man/man9f/SIZEOF_STRUCT.9f target=STRUCT_DECL.9f
480 link path=usr/share/man/man9f/STRUCT_BUF.9f target=STRUCT_DECL.9f
481 link path=usr/share/man/man9f/STRUCT_FADDR.9f target=STRUCT_DECL.9f
482 link path=usr/share/man/man9f/STRUCT_FGETP.9f target=STRUCT_DECL.9f
483 link path=usr/share/man/man9f/STRUCT_FGETP.9f target=STRUCT_DECL.9f
484 link path=usr/share/man/man9f/STRUCT_FSET.9f target=STRUCT_DECL.9f
485 link path=usr/share/man/man9f/STRUCT_FSETP.9f target=STRUCT_DECL.9f
486 link path=usr/share/man/man9f/STRUCT_HANDLE.9f target=STRUCT_DECL.9f
487 link path=usr/share/man/man9f/STRUCT_INIT.9f target=STRUCT_DECL.9f
488 link path=usr/share/man/man9f/STRUCT_SET_HANDLE.9f target=STRUCT_DECL.9f
489 link path=usr/share/man/man9f/STRUCT_SIZE.9f target=STRUCT_DECL.9f
490 link path=usr/share/man/man9f/assert.9f target=ASSERT.9f
491 link path=usr/share/man/man9f/atomic_add_16.9f target=atomic_add.9f
492 link path=usr/share/man/man9f/atomic_add_16_nv.9f target=atomic_add.9f
493 link path=usr/share/man/man9f/atomic_add_32.9f target=atomic_add.9f
494 link path=usr/share/man/man9f/atomic_add_32_nv.9f target=atomic_add.9f
495 link path=usr/share/man/man9f/atomic_add_64.9f target=atomic_add.9f
496 link path=usr/share/man/man9f/atomic_add_64_nv.9f target=atomic_add.9f
497 link path=usr/share/man/man9f/atomic_add_8.9f target=atomic_add.9f
498 link path=usr/share/man/man9f/atomic_add_8_nv.9f target=atomic_add.9f
499 link path=usr/share/man/man9f/atomic_add_char.9f target=atomic_add.9f
500 link path=usr/share/man/man9f/atomic_add_char_nv.9f target=atomic_add.9f
501 link path=usr/share/man/man9f/atomic_add_int.9f target=atomic_add.9f
502 link path=usr/share/man/man9f/atomic_add_int_nv.9f target=atomic_add.9f
503 link path=usr/share/man/man9f/atomic_add_long.9f target=atomic_add.9f
504 link path=usr/share/man/man9f/atomic_add_long_nv.9f target=atomic_add.9f
505 link path=usr/share/man/man9f/atomic_add_ptr.9f target=atomic_add.9f
506 link path=usr/share/man/man9f/atomic_add_ptr_nv.9f target=atomic_add.9f
507 link path=usr/share/man/man9f/atomic_add_short.9f target=atomic_add.9f
508 link path=usr/share/man/man9f/atomic_add_short_nv.9f target=atomic_add.9f
509 link path=usr/share/man/man9f/atomic_and_16.9f target=atomic_and.9f
510 link path=usr/share/man/man9f/atomic_and_16_nv.9f target=atomic_and.9f
511 link path=usr/share/man/man9f/atomic_and_32.9f target=atomic_and.9f
512 link path=usr/share/man/man9f/atomic_and_32_nv.9f target=atomic_and.9f
513 link path=usr/share/man/man9f/atomic_and_64.9f target=atomic_and.9f
514 link path=usr/share/man/man9f/atomic_and_64_nv.9f target=atomic_and.9f
515 link path=usr/share/man/man9f/atomic_and_8.9f target=atomic_and.9f
516 link path=usr/share/man/man9f/atomic_and_8_nv.9f target=atomic_and.9f

```

```

517 link path=usr/share/man/man9f/atomic_and_uchar.9f target=atomic_and.9f
518 link path=usr/share/man/man9f/atomic_and_uchar_nv.9f target=atomic_and.9f
519 link path=usr/share/man/man9f/atomic_and_uint.9f target=atomic_and.9f
520 link path=usr/share/man/man9f/atomic_and_uint_nv.9f target=atomic_and.9f
521 link path=usr/share/man/man9f/atomic_and_ulong.9f target=atomic_and.9f
522 link path=usr/share/man/man9f/atomic_and_ulong_nv.9f target=atomic_and.9f
523 link path=usr/share/man/man9f/atomic_and_ushort.9f target=atomic_and.9f
524 link path=usr/share/man/man9f/atomic_and_ushort_nv.9f target=atomic_and.9f
525 link path=usr/share/man/man9f/atomic_cas_16.9f target=atomic_cas.9f
526 link path=usr/share/man/man9f/atomic_cas_32.9f target=atomic_cas.9f
527 link path=usr/share/man/man9f/atomic_cas_64.9f target=atomic_cas.9f
528 link path=usr/share/man/man9f/atomic_cas_8.9f target=atomic_cas.9f
529 link path=usr/share/man/man9f/atomic_cas_ptr.9f target=atomic_cas.9f
530 link path=usr/share/man/man9f/atomic_cas_uchar.9f target=atomic_cas.9f
531 link path=usr/share/man/man9f/atomic_cas_uint.9f target=atomic_cas.9f
532 link path=usr/share/man/man9f/atomic_cas_ulong.9f target=atomic_cas.9f
533 link path=usr/share/man/man9f/atomic_cas_ushort.9f target=atomic_cas.9f
534 link path=usr/share/man/man9f/atomic_clear_long_excl.9f target=atomic_bits.9f
535 link path=usr/share/man/man9f/atomic_dec_16.9f target=atomic_dec.9f
536 link path=usr/share/man/man9f/atomic_dec_16_nv.9f target=atomic_dec.9f
537 link path=usr/share/man/man9f/atomic_dec_32.9f target=atomic_dec.9f
538 link path=usr/share/man/man9f/atomic_dec_32_nv.9f target=atomic_dec.9f
539 link path=usr/share/man/man9f/atomic_dec_64.9f target=atomic_dec.9f
540 link path=usr/share/man/man9f/atomic_dec_64_nv.9f target=atomic_dec.9f
541 link path=usr/share/man/man9f/atomic_dec_8.9f target=atomic_dec.9f
542 link path=usr/share/man/man9f/atomic_dec_8_nv.9f target=atomic_dec.9f
543 link path=usr/share/man/man9f/atomic_dec_ptr.9f target=atomic_dec.9f
544 link path=usr/share/man/man9f/atomic_dec_ptr_nv.9f target=atomic_dec.9f
545 link path=usr/share/man/man9f/atomic_dec_uchar.9f target=atomic_dec.9f
546 link path=usr/share/man/man9f/atomic_dec_uchar_nv.9f target=atomic_dec.9f
547 link path=usr/share/man/man9f/atomic_dec_uint.9f target=atomic_dec.9f
548 link path=usr/share/man/man9f/atomic_dec_uint_nv.9f target=atomic_dec.9f
549 link path=usr/share/man/man9f/atomic_dec_ulong.9f target=atomic_dec.9f
550 link path=usr/share/man/man9f/atomic_dec_ulong_nv.9f target=atomic_dec.9f
551 link path=usr/share/man/man9f/atomic_dec_ushort.9f target=atomic_dec.9f
552 link path=usr/share/man/man9f/atomic_dec_ushort_nv.9f target=atomic_dec.9f
553 link path=usr/share/man/man9f/atomic_inc_16.9f target=atomic_inc.9f
554 link path=usr/share/man/man9f/atomic_inc_16_nv.9f target=atomic_inc.9f
555 link path=usr/share/man/man9f/atomic_inc_32.9f target=atomic_inc.9f
556 link path=usr/share/man/man9f/atomic_inc_32_nv.9f target=atomic_inc.9f
557 link path=usr/share/man/man9f/atomic_inc_64.9f target=atomic_inc.9f
558 link path=usr/share/man/man9f/atomic_inc_64_nv.9f target=atomic_inc.9f
559 link path=usr/share/man/man9f/atomic_inc_8.9f target=atomic_inc.9f
560 link path=usr/share/man/man9f/atomic_inc_8_nv.9f target=atomic_inc.9f
561 link path=usr/share/man/man9f/atomic_inc_ptr.9f target=atomic_inc.9f
562 link path=usr/share/man/man9f/atomic_inc_ptr_nv.9f target=atomic_inc.9f
563 link path=usr/share/man/man9f/atomic_inc_uchar.9f target=atomic_inc.9f
564 link path=usr/share/man/man9f/atomic_inc_uchar_nv.9f target=atomic_inc.9f
565 link path=usr/share/man/man9f/atomic_inc_uint.9f target=atomic_inc.9f
566 link path=usr/share/man/man9f/atomic_inc_uint_nv.9f target=atomic_inc.9f
567 link path=usr/share/man/man9f/atomic_inc_ulong.9f target=atomic_inc.9f
568 link path=usr/share/man/man9f/atomic_inc_ulong_nv.9f target=atomic_inc.9f
569 link path=usr/share/man/man9f/atomic_inc_ushort.9f target=atomic_inc.9f
570 link path=usr/share/man/man9f/atomic_inc_ushort_nv.9f target=atomic_inc.9f
571 link path=usr/share/man/man9f/atomic_or_16.9f target=atomic_or.9f
572 link path=usr/share/man/man9f/atomic_or_16_nv.9f target=atomic_or.9f
573 link path=usr/share/man/man9f/atomic_or_32.9f target=atomic_or.9f
574 link path=usr/share/man/man9f/atomic_or_32_nv.9f target=atomic_or.9f
575 link path=usr/share/man/man9f/atomic_or_64.9f target=atomic_or.9f
576 link path=usr/share/man/man9f/atomic_or_64_nv.9f target=atomic_or.9f
577 link path=usr/share/man/man9f/atomic_or_8.9f target=atomic_or.9f
578 link path=usr/share/man/man9f/atomic_or_8_nv.9f target=atomic_or.9f
579 link path=usr/share/man/man9f/atomic_or_uchar.9f target=atomic_or.9f
580 link path=usr/share/man/man9f/atomic_or_uchar_nv.9f target=atomic_or.9f
581 link path=usr/share/man/man9f/atomic_or_uint.9f target=atomic_or.9f
582 link path=usr/share/man/man9f/atomic_or_uint_nv.9f target=atomic_or.9f

```

```

583 link path=usr/share/man/man9f/atomic_or_ulong.9f target=atomic_or.9f
584 link path=usr/share/man/man9f/atomic_or_ulong_nv.9f target=atomic_or.9f
585 link path=usr/share/man/man9f/atomic_or_ushort.9f target=atomic_or.9f
586 link path=usr/share/man/man9f/atomic_or_ushort_nv.9f target=atomic_or.9f
587 link path=usr/share/man/man9f/atomic_set_long_excl.9f target=atomic_bits.9f
588 link path=usr/share/man/man9f/atomic_swap_16.9f target=atomic_swap.9f
589 link path=usr/share/man/man9f/atomic_swap_32.9f target=atomic_swap.9f
590 link path=usr/share/man/man9f/atomic_swap_64.9f target=atomic_swap.9f
591 link path=usr/share/man/man9f/atomic_swap_8.9f target=atomic_swap.9f
592 link path=usr/share/man/man9f/atomic_swap_ptr.9f target=atomic_swap.9f
593 link path=usr/share/man/man9f/atomic_swap_uchar.9f target=atomic_swap.9f
594 link path=usr/share/man/man9f/atomic_swap_uint.9f target=atomic_swap.9f
595 link path=usr/share/man/man9f/atomic_swap_ulong.9f target=atomic_swap.9f
596 link path=usr/share/man/man9f/atomic_swap_ushort.9f target=atomic_swap.9f
597 link path=usr/share/man/man9f/crgetgid.9f target=ddi_cred.9f
598 link path=usr/share/man/man9f/crgetgroups.9f target=ddi_cred.9f
599 link path=usr/share/man/man9f/crgetngroups.9f target=ddi_cred.9f
600 link path=usr/share/man/man9f/crgetrgid.9f target=ddi_cred.9f
601 link path=usr/share/man/man9f/crgetruid.9f target=ddi_cred.9f
602 link path=usr/share/man/man9f/crgetsgid.9f target=ddi_cred.9f
603 link path=usr/share/man/man9f/crgetsuid.9f target=ddi_cred.9f
604 link path=usr/share/man/man9f/crgetuid.9f target=ddi_cred.9f
605 link path=usr/share/man/man9f/crgetzoneid.9f target=ddi_cred.9f
606 link path=usr/share/man/man9f/csx_Get16.9f target=csx_Get8.9f
607 link path=usr/share/man/man9f/csx_Get32.9f target=csx_Get8.9f
608 link path=usr/share/man/man9f/csx_Get64.9f target=csx_Get8.9f
609 link path=usr/share/man/man9f/csx_GetEventMask.9f target=csx_SetEventMask.9f
610 link path=usr/share/man/man9f/csx_GetNextClient.9f \
611 target=csx_GetFirstClient.9f
612 link path=usr/share/man/man9f/csx_GetNextTuple.9f target=csx_GetFirstTuple.9f
613 link path=usr/share/man/man9f/csx_Parse_CISTPL_DEVICE_A.9f \
614 target=csx_Parse_CISTPL_DEVICE.9f
615 link path=usr/share/man/man9f/csx_Parse_CISTPL_DEVICE_OA.9f \
616 target=csx_Parse_CISTPL_DEVICE.9f
617 link path=usr/share/man/man9f/csx_Parse_CISTPL_DEVICE_OC.9f \
618 target=csx_Parse_CISTPL_DEVICE.9f
619 link path=usr/share/man/man9f/csx_Parse_CISTPL_JEDEC_A.9f \
620 target=csx_Parse_CISTPL_JEDEC_C.9f
621 link path=usr/share/man/man9f/csx_Parse_CISTPL_LONGLINK_C.9f \
622 target=csx_Parse_CISTPL_LONGLINK_A.9f
623 link path=usr/share/man/man9f/csx_Put16.9f target=csx_Put8.9f
624 link path=usr/share/man/man9f/csx_Put32.9f target=csx_Put8.9f
625 link path=usr/share/man/man9f/csx_Put64.9f target=csx_Put8.9f
626 link path=usr/share/man/man9f/csx_ReleaseIO.9f target=csx_RequestIO.9f
627 link path=usr/share/man/man9f/csx_ReleaseIRQ.9f target=csx_RequestIRQ.9f
628 link path=usr/share/man/man9f/csx_ReleaseSocketMask.9f \
629 target=csx_RequestSocketMask.9f
630 link path=usr/share/man/man9f/csx_ReleaseWindow.9f target=csx_RequestWindow.9f
631 link path=usr/share/man/man9f/csx_RemoveDeviceNode.9f \
632 target=csx_MakeDeviceNode.9f
633 link path=usr/share/man/man9f/csx_RepGet16.9f target=csx_RepGet8.9f
634 link path=usr/share/man/man9f/csx_RepGet32.9f target=csx_RepGet8.9f
635 link path=usr/share/man/man9f/csx_RepGet64.9f target=csx_RepGet8.9f
636 link path=usr/share/man/man9f/csx_RepPut16.9f target=csx_RepPut8.9f
637 link path=usr/share/man/man9f/csx_RepPut32.9f target=csx_RepPut8.9f
638 link path=usr/share/man/man9f/csx_RepPut64.9f target=csx_RepPut8.9f
639 link path=usr/share/man/man9f/cv_broadcast.9f target=condvar.9f
640 link path=usr/share/man/man9f/cv_destroy.9f target=condvar.9f
641 link path=usr/share/man/man9f/cv_init.9f target=condvar.9f
642 link path=usr/share/man/man9f/cv_reltimedwait.9f target=condvar.9f
643 link path=usr/share/man/man9f/cv_reltimedwait_sig.9f target=condvar.9f
644 link path=usr/share/man/man9f/cv_signal.9f target=condvar.9f
645 link path=usr/share/man/man9f/cv_timedwait.9f target=condvar.9f
646 link path=usr/share/man/man9f/cv_timedwait_sig.9f target=condvar.9f
647 link path=usr/share/man/man9f/cv_wait.9f target=condvar.9f
648 link path=usr/share/man/man9f/cv_wait_sig.9f target=condvar.9f

```

```

649 link path=usr/share/man/man9f/ddi_btopr.9f target=ddi_btop.9f
650 link path=usr/share/man/man9f/ddi_cb_unregister.9f target=ddi_cb_register.9f
651 link path=usr/share/man/man9f/ddi_check_dma_handle.9f \
652 target=ddi_check_acc_handle.9f
653 link path=usr/share/man/man9f/ddi_devid_free.9f target=ddi_devid_compare.9f
654 link path=usr/share/man/man9f/ddi_devid_get.9f target=ddi_devid_compare.9f
655 link path=usr/share/man/man9f/ddi_devid_init.9f target=ddi_devid_compare.9f
656 link path=usr/share/man/man9f/ddi_devid_register.9f \
657 target=ddi_devid_compare.9f
658 link path=usr/share/man/man9f/ddi_devid_sizeof.9f target=ddi_devid_compare.9f
659 link path=usr/share/man/man9f/ddi_devid_str_decode.9f \
660 target=ddi_devid_compare.9f
661 link path=usr/share/man/man9f/ddi_devid_str_encode.9f \
662 target=ddi_devid_compare.9f
663 link path=usr/share/man/man9f/ddi_devid_str_free.9f \
664 target=ddi_devid_compare.9f
665 link path=usr/share/man/man9f/ddi_devid_unregister.9f \
666 target=ddi_devid_compare.9f
667 link path=usr/share/man/man9f/ddi_devid_valid.9f target=ddi_devid_compare.9f
668 link path=usr/share/man/man9f/ddi_devmap_segmap.9f target=devmap_setup.9f
669 link path=usr/share/man/man9f/ddi_dmae_lstparty.9f target=ddi_dmae.9f
670 link path=usr/share/man/man9f/ddi_dmae_alloc.9f target=ddi_dmae.9f
671 link path=usr/share/man/man9f/ddi_dmae_disable.9f target=ddi_dmae.9f
672 link path=usr/share/man/man9f/ddi_dmae_enable.9f target=ddi_dmae.9f
673 link path=usr/share/man/man9f/ddi_dmae_getattr.9f target=ddi_dmae.9f
674 link path=usr/share/man/man9f/ddi_dmae_getcnt.9f target=ddi_dmae.9f
675 link path=usr/share/man/man9f/ddi_dmae_getlim.9f target=ddi_dmae.9f
676 link path=usr/share/man/man9f/ddi_dmae_prog.9f target=ddi_dmae.9f
677 link path=usr/share/man/man9f/ddi_dmae_release.9f target=ddi_dmae.9f
678 link path=usr/share/man/man9f/ddi_dmae_stop.9f target=ddi_dmae.9f
679 link path=usr/share/man/man9f/ddi_exit_critical.9f \
680 target=ddi_enter_critical.9f
681 link path=usr/share/man/man9f/ddi_ffs.9f target=ddi_ffs.9f
682 link path=usr/share/man/man9f/ddi_fm_capable.9f target=ddi_fm_init.9f
683 link path=usr/share/man/man9f/ddi_fm_dma_err_clear.9f \
684 target=ddi_fm_acc_err_clear.9f
685 link path=usr/share/man/man9f/ddi_fm_dma_err_get.9f \
686 target=ddi_fm_acc_err_get.9f
687 link path=usr/share/man/man9f/ddi_fm_fini.9f target=ddi_fm_init.9f
688 link path=usr/share/man/man9f/ddi_fm_handler_unregister.9f \
689 target=ddi_fm_handler_register.9f
690 link path=usr/share/man/man9f/ddi_get16.9f target=ddi_get8.9f
691 link path=usr/share/man/man9f/ddi_get32.9f target=ddi_get8.9f
692 link path=usr/share/man/man9f/ddi_get64.9f target=ddi_get8.9f
693 link path=usr/share/man/man9f/ddi_get_iblock_cookie.9f target=ddi_add_intr.9f
694 link path=usr/share/man/man9f/ddi_get_lbolt64.9f target=ddi_get_lbolt.9f
695 link path=usr/share/man/man9f/ddi_get_name.9f target=ddi_binding_name.9f
696 link path=usr/share/man/man9f/ddi_get_soft_iblock_cookie.9f \
697 target=ddi_add_softintr.9f
698 link path=usr/share/man/man9f/ddi_get_soft_state.9f target=ddi_soft_state.9f
699 link path=usr/share/man/man9f/ddi_getb.9f target=ddi_get8.9f
700 link path=usr/share/man/man9f/ddi_getl.9f target=ddi_get8.9f
701 link path=usr/share/man/man9f/ddi_getll.9f target=ddi_get8.9f
702 link path=usr/share/man/man9f/ddi_getlongprop.9f target=ddi_prop_op.9f
703 link path=usr/share/man/man9f/ddi_getlongprop_buf.9f target=ddi_prop_op.9f
704 link path=usr/share/man/man9f/ddi_getprop.9f target=ddi_prop_op.9f
705 link path=usr/share/man/man9f/ddi_getpropLen.9f target=ddi_prop_op.9f
706 link path=usr/share/man/man9f/ddi_getw.9f target=ddi_get8.9f
707 link path=usr/share/man/man9f/ddi_intr_block_disable.9f \
708 target=ddi_intr_enable.9f
709 link path=usr/share/man/man9f/ddi_intr_block_enable.9f \
710 target=ddi_intr_enable.9f
711 link path=usr/share/man/man9f/ddi_intr_clr_mask.9f target=ddi_intr_set_mask.9f
712 link path=usr/share/man/man9f/ddi_intr_disable.9f target=ddi_intr_enable.9f
713 link path=usr/share/man/man9f/ddi_intr_free.9f target=ddi_intr_alloc.9f
714 link path=usr/share/man/man9f/ddi_intr_get_navail.9f \

```

```

714 target=ddi_intr_get_nintrs.9f
715 link path=usr/share/man/man9f/ddi_intr_get_softint_pri.9f \
716 target=ddi_intr_add_softint.9f
717 link path=usr/share/man/man9f/ddi_intr_remove_handler.9f \
718 target=ddi_intr_add_handler.9f
719 link path=usr/share/man/man9f/ddi_intr_remove_softint.9f \
720 target=ddi_intr_add_softint.9f
721 link path=usr/share/man/man9f/ddi_intr_set_cap.9f target=ddi_intr_get_cap.9f
722 link path=usr/share/man/man9f/ddi_intr_set_pri.9f target=ddi_intr_get_pri.9f
723 link path=usr/share/man/man9f/ddi_intr_set_softint_pri.9f \
724 target=ddi_intr_add_softint.9f
725 link path=usr/share/man/man9f/ddi_intr_trigger_softint.9f \
726 target=ddi_intr_add_softint.9f
727 link path=usr/share/man/man9f/ddi_io_get16.9f target=ddi_io_get8.9f
728 link path=usr/share/man/man9f/ddi_io_get32.9f target=ddi_io_get8.9f
729 link path=usr/share/man/man9f/ddi_io_getb.9f target=ddi_io_get8.9f
730 link path=usr/share/man/man9f/ddi_io_getl.9f target=ddi_io_get8.9f
731 link path=usr/share/man/man9f/ddi_io_getw.9f target=ddi_io_get8.9f
732 link path=usr/share/man/man9f/ddi_io_put16.9f target=ddi_io_put8.9f
733 link path=usr/share/man/man9f/ddi_io_put32.9f target=ddi_io_put8.9f
734 link path=usr/share/man/man9f/ddi_io_putb.9f target=ddi_io_put8.9f
735 link path=usr/share/man/man9f/ddi_io_putl.9f target=ddi_io_put8.9f
736 link path=usr/share/man/man9f/ddi_io_putw.9f target=ddi_io_put8.9f
737 link path=usr/share/man/man9f/ddi_io_rep_get16.9f target=ddi_io_rep_get8.9f
738 link path=usr/share/man/man9f/ddi_io_rep_get32.9f target=ddi_io_rep_get8.9f
739 link path=usr/share/man/man9f/ddi_io_rep_getb.9f target=ddi_io_rep_get8.9f
740 link path=usr/share/man/man9f/ddi_io_rep_getl.9f target=ddi_io_rep_get8.9f
741 link path=usr/share/man/man9f/ddi_io_rep_getw.9f target=ddi_io_rep_get8.9f
742 link path=usr/share/man/man9f/ddi_io_rep_put16.9f target=ddi_io_rep_put8.9f
743 link path=usr/share/man/man9f/ddi_io_rep_put32.9f target=ddi_io_rep_put8.9f
744 link path=usr/share/man/man9f/ddi_io_rep_putb.9f target=ddi_io_rep_put8.9f
745 link path=usr/share/man/man9f/ddi_io_rep_putl.9f target=ddi_io_rep_put8.9f
746 link path=usr/share/man/man9f/ddi_io_rep_putw.9f target=ddi_io_rep_put8.9f
747 link path=usr/share/man/man9f/ddi_mem_get16.9f target=ddi_mem_get8.9f
748 link path=usr/share/man/man9f/ddi_mem_get32.9f target=ddi_mem_get8.9f
749 link path=usr/share/man/man9f/ddi_mem_get64.9f target=ddi_mem_get8.9f
750 link path=usr/share/man/man9f/ddi_mem_getb.9f target=ddi_mem_get8.9f
751 link path=usr/share/man/man9f/ddi_mem_getl.9f target=ddi_mem_get8.9f
752 link path=usr/share/man/man9f/ddi_mem_getll.9f target=ddi_mem_get8.9f
753 link path=usr/share/man/man9f/ddi_mem_getw.9f target=ddi_mem_get8.9f
754 link path=usr/share/man/man9f/ddi_mem_put16.9f target=ddi_mem_put8.9f
755 link path=usr/share/man/man9f/ddi_mem_put32.9f target=ddi_mem_put8.9f
756 link path=usr/share/man/man9f/ddi_mem_put64.9f target=ddi_mem_put8.9f
757 link path=usr/share/man/man9f/ddi_mem_putb.9f target=ddi_mem_put8.9f
758 link path=usr/share/man/man9f/ddi_mem_putl.9f target=ddi_mem_put8.9f
759 link path=usr/share/man/man9f/ddi_mem_putll.9f target=ddi_mem_put8.9f
760 link path=usr/share/man/man9f/ddi_mem_putw.9f target=ddi_mem_put8.9f
761 link path=usr/share/man/man9f/ddi_mem_rep_get16.9f target=ddi_mem_rep_get8.9f
762 link path=usr/share/man/man9f/ddi_mem_rep_get32.9f target=ddi_mem_rep_get8.9f
763 link path=usr/share/man/man9f/ddi_mem_rep_get64.9f target=ddi_mem_rep_get8.9f
764 link path=usr/share/man/man9f/ddi_mem_rep_getb.9f target=ddi_mem_rep_get8.9f
765 link path=usr/share/man/man9f/ddi_mem_rep_getl.9f target=ddi_mem_rep_get8.9f
766 link path=usr/share/man/man9f/ddi_mem_rep_getll.9f target=ddi_mem_rep_get8.9f
767 link path=usr/share/man/man9f/ddi_mem_rep_getw.9f target=ddi_mem_rep_get8.9f
768 link path=usr/share/man/man9f/ddi_mem_rep_put16.9f target=ddi_mem_rep_put8.9f
769 link path=usr/share/man/man9f/ddi_mem_rep_put32.9f target=ddi_mem_rep_put8.9f
770 link path=usr/share/man/man9f/ddi_mem_rep_put64.9f target=ddi_mem_rep_put8.9f
771 link path=usr/share/man/man9f/ddi_mem_rep_putb.9f target=ddi_mem_rep_put8.9f
772 link path=usr/share/man/man9f/ddi_mem_rep_putl.9f target=ddi_mem_rep_put8.9f
773 link path=usr/share/man/man9f/ddi_mem_rep_putll.9f target=ddi_mem_rep_put8.9f
774 link path=usr/share/man/man9f/ddi_mem_rep_putw.9f target=ddi_mem_rep_put8.9f
775 link path=usr/share/man/man9f/ddi_modclose.9f target=ddi_modopen.9f
776 link path=usr/share/man/man9f/ddi_modsym.9f target=ddi_modopen.9f
777 link path=usr/share/man/man9f/ddi_peek16.9f target=ddi_peek.9f
778 link path=usr/share/man/man9f/ddi_peek32.9f target=ddi_peek.9f
779 link path=usr/share/man/man9f/ddi_peek64.9f target=ddi_peek.9f

```

```

780 link path=usr/share/man/man9f/ddi_peek8.9f target=ddi_peek.9f
781 link path=usr/share/man/man9f/ddi_peekc.9f target=ddi_peek.9f
782 link path=usr/share/man/man9f/ddi_peekd.9f target=ddi_peek.9f
783 link path=usr/share/man/man9f/ddi_peekl.9f target=ddi_peek.9f
784 link path=usr/share/man/man9f/ddi_peeks.9f target=ddi_peek.9f
785 link path=usr/share/man/man9f/ddi_poke16.9f target=ddi_poke.9f
786 link path=usr/share/man/man9f/ddi_poke32.9f target=ddi_poke.9f
787 link path=usr/share/man/man9f/ddi_poke64.9f target=ddi_poke.9f
788 link path=usr/share/man/man9f/ddi_poke8.9f target=ddi_poke.9f
789 link path=usr/share/man/man9f/ddi_pokec.9f target=ddi_poke.9f
790 link path=usr/share/man/man9f/ddi_poked.9f target=ddi_poke.9f
791 link path=usr/share/man/man9f/ddi_pokel.9f target=ddi_poke.9f
792 link path=usr/share/man/man9f/ddi_pokes.9f target=ddi_poke.9f
793 link path=usr/share/man/man9f/ddi_prop_free.9f target=ddi_prop_lookup.9f
794 link path=usr/share/man/man9f/ddi_prop_get_int64.9f target=ddi_prop_get_int.9f
795 link path=usr/share/man/man9f/ddi_prop_lookup_byte_array.9f \
796 target=ddi_prop_lookup.9f
797 link path=usr/share/man/man9f/ddi_prop_lookup_int64_array.9f \
798 target=ddi_prop_lookup.9f
799 link path=usr/share/man/man9f/ddi_prop_lookup_int_array.9f \
800 target=ddi_prop_lookup.9f
801 link path=usr/share/man/man9f/ddi_prop_lookup_string.9f \
802 target=ddi_prop_lookup.9f
803 link path=usr/share/man/man9f/ddi_prop_lookup_string_array.9f \
804 target=ddi_prop_lookup.9f
805 link path=usr/share/man/man9f/ddi_prop_modify.9f target=ddi_prop_create.9f
806 link path=usr/share/man/man9f/ddi_prop_remove.9f target=ddi_prop_create.9f
807 link path=usr/share/man/man9f/ddi_prop_remove_all.9f target=ddi_prop_create.9f
808 link path=usr/share/man/man9f/ddi_prop_undefine.9f target=ddi_prop_create.9f
809 link path=usr/share/man/man9f/ddi_prop_update_byte_array.9f \
810 target=ddi_prop_update.9f
811 link path=usr/share/man/man9f/ddi_prop_update_int.9f target=ddi_prop_update.9f
812 link path=usr/share/man/man9f/ddi_prop_update_int64.9f \
813 target=ddi_prop_update.9f
814 link path=usr/share/man/man9f/ddi_prop_update_int64_array.9f \
815 target=ddi_prop_update.9f
816 link path=usr/share/man/man9f/ddi_prop_update_int_array.9f \
817 target=ddi_prop_update.9f
818 link path=usr/share/man/man9f/ddi_prop_update_string.9f \
819 target=ddi_prop_update.9f
820 link path=usr/share/man/man9f/ddi_prop_update_string_array.9f \
821 target=ddi_prop_update.9f
822 link path=usr/share/man/man9f/ddi_ptob.9f target=ddi_btob.9f
823 link path=usr/share/man/man9f/ddi_put16.9f target=ddi_put8.9f
824 link path=usr/share/man/man9f/ddi_put32.9f target=ddi_put8.9f
825 link path=usr/share/man/man9f/ddi_put64.9f target=ddi_put8.9f
826 link path=usr/share/man/man9f/ddi_putb.9f target=ddi_put8.9f
827 link path=usr/share/man/man9f/ddi_putl.9f target=ddi_put8.9f
828 link path=usr/share/man/man9f/ddi_putll.9f target=ddi_put8.9f
829 link path=usr/share/man/man9f/ddi_putw.9f target=ddi_put8.9f
830 link path=usr/share/man/man9f/ddi_remove_intr.9f target=ddi_add_intr.9f
831 link path=usr/share/man/man9f/ddi_remove_softintr.9f \
832 target=ddi_add_softintr.9f
833 link path=usr/share/man/man9f/ddi_rep_get16.9f target=ddi_rep_get8.9f
834 link path=usr/share/man/man9f/ddi_rep_get32.9f target=ddi_rep_get8.9f
835 link path=usr/share/man/man9f/ddi_rep_get64.9f target=ddi_rep_get8.9f
836 link path=usr/share/man/man9f/ddi_rep_getb.9f target=ddi_rep_get8.9f
837 link path=usr/share/man/man9f/ddi_rep_getl.9f target=ddi_rep_get8.9f
838 link path=usr/share/man/man9f/ddi_rep_getll.9f target=ddi_rep_get8.9f
839 link path=usr/share/man/man9f/ddi_rep_getw.9f target=ddi_rep_get8.9f
840 link path=usr/share/man/man9f/ddi_rep_put16.9f target=ddi_rep_put8.9f
841 link path=usr/share/man/man9f/ddi_rep_put32.9f target=ddi_rep_put8.9f
842 link path=usr/share/man/man9f/ddi_rep_put64.9f target=ddi_rep_put8.9f
843 link path=usr/share/man/man9f/ddi_rep_putb.9f target=ddi_rep_put8.9f
844 link path=usr/share/man/man9f/ddi_rep_putl.9f target=ddi_rep_put8.9f
845 link path=usr/share/man/man9f/ddi_rep_putll.9f target=ddi_rep_put8.9f

```

```

846 link path=usr/share/man/man9f/ddi_rep_putw.9f target=ddi_rep_put8.9f
847 link path=usr/share/man/man9f/ddi_segmap_setup.9f target=ddi_segmap.9f
848 link path=usr/share/man/man9f/ddi_set_driver_private.9f \
849 target=ddi_get_driver_private.9f
850 link path=usr/share/man/man9f/ddi_soft_state_fini.9f target=ddi_soft_state.9f
851 link path=usr/share/man/man9f/ddi_soft_state_free.9f target=ddi_soft_state.9f
852 link path=usr/share/man/man9f/ddi_soft_state_init.9f target=ddi_soft_state.9f
853 link path=usr/share/man/man9f/ddi_soft_state_zalloc.9f \
854 target=ddi_soft_state.9f
855 link path=usr/share/man/man9f/ddi_strdup.9f target=string.9f
856 link path=usr/share/man/man9f/ddi_strtoul.9f target=ddi_strtoll.9f
857 link path=usr/share/man/man9f/ddi_taskq_create.9f target=taskq.9f
858 link path=usr/share/man/man9f/ddi_taskq_destroy.9f target=taskq.9f
859 link path=usr/share/man/man9f/ddi_taskq_dispatch.9f target=taskq.9f
860 link path=usr/share/man/man9f/ddi_taskq_resume.9f target=taskq.9f
861 link path=usr/share/man/man9f/ddi_taskq_suspend.9f target=taskq.9f
862 link path=usr/share/man/man9f/ddi_taskq_wait.9f target=taskq.9f
863 link path=usr/share/man/man9f/ddi_trigger_softintr.9f \
864 target=ddi_add_softintr.9f
865 link path=usr/share/man/man9f/ddi_umem_free.9f target=ddi_umem_alloc.9f
866 link path=usr/share/man/man9f/ddi_umem_unlock.9f target=ddi_umem_lock.9f
867 link path=usr/share/man/man9f/ddi_unmap_regs.9f target=ddi_map_regs.9f
868 link path=usr/share/man/man9f/desballoc.9f target=esballoc.9f
869 link path=usr/share/man/man9f/devmap_load.9f target=devmap_unload.9f
870 link path=usr/share/man/man9f/devmap_umem_setup.9f \
871 target=devmap_devmem_setup.9f
872 link path=usr/share/man/man9f/dlerrorack.9f target=dlbindack.9f
873 link path=usr/share/man/man9f/dlokack.9f target=dlbindack.9f
874 link path=usr/share/man/man9f/dlphysaddrack.9f target=dlbindack.9f
875 link path=usr/share/man/man9f/dluderrorind.9f target=dlbindack.9f
876 link path=usr/share/man/man9f/free_pktiopb.9f target=get_pktiopb.9f
877 link path=usr/share/man/man9f/gld_intr.9f target=gld.9f
878 link path=usr/share/man/man9f/gld_mac_alloc.9f target=gld.9f
879 link path=usr/share/man/man9f/gld_mac_free.9f target=gld.9f
880 link path=usr/share/man/man9f/gld_recv.9f target=gld.9f
881 link path=usr/share/man/man9f/gld_register.9f target=gld.9f
882 link path=usr/share/man/man9f/gld_sched.9f target=gld.9f
883 link path=usr/share/man/man9f/gld_unregister.9f target=gld.9f
884 link path=usr/share/man/man9f/id32_free.9f target=id32_alloc.9f
885 link path=usr/share/man/man9f/id32_lookup.9f target=id32_alloc.9f
886 link path=usr/share/man/man9f/inl.9f target=inb.9f
887 link path=usr/share/man/man9f/intro.9f target=intro.9f
888 link path=usr/share/man/man9f/inw.9f target=inb.9f
889 link path=usr/share/man/man9f/kmem_cache_alloc.9f target=kmem_cache_create.9f
890 link path=usr/share/man/man9f/kmem_cache_destroy.9f \
891 target=kmem_cache_create.9f
892 link path=usr/share/man/man9f/kmem_cache_free.9f target=kmem_cache_create.9f
893 link path=usr/share/man/man9f/kmem_cache_set_move.9f \
894 target=kmem_cache_create.9f
895 link path=usr/share/man/man9f/kmem_free.9f target=kmem_alloc.9f
896 link path=usr/share/man/man9f/kmem_zalloc.9f target=kmem_alloc.9f
897 link path=usr/share/man/man9f/kstat_named_setstr.9f target=kstat_named_init.9f
898 link path=usr/share/man/man9f/kstat_runq_back_to_waitq.9f \
899 target=kstat_queue.9f
900 link path=usr/share/man/man9f/kstat_runq_enter.9f target=kstat_queue.9f
901 link path=usr/share/man/man9f/kstat_runq_exit.9f target=kstat_queue.9f
902 link path=usr/share/man/man9f/kstat_waitq_enter.9f target=kstat_queue.9f
903 link path=usr/share/man/man9f/kstat_waitq_exit.9f target=kstat_queue.9f
904 link path=usr/share/man/man9f/kstat_waitq_to_runq.9f target=kstat_queue.9f
905 link path=usr/share/man/man9f/ldi_await.9f target=ldi_await.9f
906 link path=usr/share/man/man9f/ldi_close.9f target=ldi_open_by_dev.9f
907 link path=usr/share/man/man9f/ldi_get_devid.9f target=ldi_get_dev.9f
908 link path=usr/share/man/man9f/ldi_get_minor_name.9f target=ldi_get_dev.9f
909 link path=usr/share/man/man9f/ldi_get_otyp.9f target=ldi_get_dev.9f
910 link path=usr/share/man/man9f/ldi_getmsg.9f target=ldi_putmsg.9f
911 link path=usr/share/man/man9f/ldi_ident_from_dip.9f \

```

```

912     target=ldi_ident_from_dev.9f
913 link path=usr/share/man/man9f/ldi_ident_from_stream.9f \
914     target=ldi_ident_from_dev.9f
915 link path=usr/share/man/man9f/ldi_ident_release.9f \
916     target=ldi_ident_from_dev.9f
917 link path=usr/share/man/man9f/ldi_open_by_devid.9f target=ldi_open_by_dev.9f
918 link path=usr/share/man/man9f/ldi_open_by_name.9f target=ldi_open_by_dev.9f
919 link path=usr/share/man/man9f/ldi_prop_get_int64.9f target=ldi_prop_get_int.9f
920 link path=usr/share/man/man9f/ldi_prop_lookup_byte_array.9f \
921     target=ldi_prop_lookup_int_array.9f
922 link path=usr/share/man/man9f/ldi_prop_lookup_int64_array.9f \
923     target=ldi_prop_lookup_int_array.9f
924 link path=usr/share/man/man9f/ldi_prop_lookup_string.9f \
925     target=ldi_prop_lookup_int_array.9f
926 link path=usr/share/man/man9f/ldi_prop_lookup_string_array.9f \
927     target=ldi_prop_lookup_int_array.9f
928 link path=usr/share/man/man9f/ldi_write.9f target=ldi_read.9f
929 link path=usr/share/man/man9f/list_destroy.9f target=list_create.9f
930 link path=usr/share/man/man9f/list_head.9f target=list_create.9f
931 link path=usr/share/man/man9f/list_insert_after.9f target=list_create.9f
932 link path=usr/share/man/man9f/list_insert_before.9f target=list_create.9f
933 link path=usr/share/man/man9f/list_insert_head.9f target=list_create.9f
934 link path=usr/share/man/man9f/list_insert_tail.9f target=list_create.9f
935 link path=usr/share/man/man9f/list_is_empty.9f target=list_create.9f
936 link path=usr/share/man/man9f/list_link_active.9f target=list_create.9f
937 link path=usr/share/man/man9f/list_link_init.9f target=list_create.9f
938 link path=usr/share/man/man9f/list_link_replace.9f target=list_create.9f
939 link path=usr/share/man/man9f/list_move_tail.9f target=list_create.9f
940 link path=usr/share/man/man9f/list_next.9f target=list_create.9f
941 link path=usr/share/man/man9f/list_prev.9f target=list_create.9f
942 link path=usr/share/man/man9f/list_remove.9f target=list_create.9f
943 link path=usr/share/man/man9f/list_remove_head.9f target=list_create.9f
944 link path=usr/share/man/man9f/list_remove_tail.9f target=list_create.9f
945 link path=usr/share/man/man9f/list_tail.9f target=list_create.9f
946 link path=usr/share/man/man9f/makecom_g0.9f target=makecom.9f
947 link path=usr/share/man/man9f/makecom_g0_s.9f target=makecom.9f
948 link path=usr/share/man/man9f/makecom_g1.9f target=makecom.9f
949 link path=usr/share/man/man9f/makecom_g5.9f target=makecom.9f
950 link path=usr/share/man/man9f/membar_consumer.9f target=membar_ops.9f
951 link path=usr/share/man/man9f/membar_enter.9f target=membar_ops.9f
952 link path=usr/share/man/man9f/membar_exit.9f target=membar_ops.9f
953 link path=usr/share/man/man9f/membar_producer.9f target=membar_ops.9f
954 link path=usr/share/man/man9f/memcmp.9f target=memchr.9f
955 link path=usr/share/man/man9f/memcpy.9f target=memchr.9f
956 link path=usr/share/man/man9f/memmove.9f target=memchr.9f
957 link path=usr/share/man/man9f/memset.9f target=memchr.9f
958 link path=usr/share/man/man9f/minphys.9f target=physio.9f
959 link path=usr/share/man/man9f/mod_info.9f target=mod_install.9f
960 link path=usr/share/man/man9f/mod_modname.9f target=mod_install.9f
961 link path=usr/share/man/man9f/mod_remove.9f target=mod_install.9f
962 link path=usr/share/man/man9f/mutex_destroy.9f target=mutex.9f
963 link path=usr/share/man/man9f/mutex_enter.9f target=mutex.9f
964 link path=usr/share/man/man9f/mutex_exit.9f target=mutex.9f
965 link path=usr/share/man/man9f/mutex_init.9f target=mutex.9f
966 link path=usr/share/man/man9f/mutex_owned.9f target=mutex.9f
967 link path=usr/share/man/man9f/mutex_tryenter.9f target=mutex.9f
968 link path=usr/share/man/man9f/net_event_notify_unregister.9f \
969     target=net_event_notify_register.9f
970 link path=usr/share/man/man9f/net_instance_notify_unregister.9f \
971     target=net_instance_notify_register.9f
972 link path=usr/share/man/man9f/net_instance_protocol_unregister.9f \
973     target=net_protocol_notify_register.9f
974 link path=usr/share/man/man9f/numtos.9f target=stoi.9f
975 link path=usr/share/man/man9f/nv_alloc_fini.9f target=nvlist_alloc.9f
976 link path=usr/share/man/man9f/nv_alloc_init.9f target=nvlist_alloc.9f
977 link path=usr/share/man/man9f/nvlist_add_boolean_array.9f \

```

```

978     target=nvlist_add_boolean.9f
979 link path=usr/share/man/man9f/nvlist_add_boolean_value.9f \
980     target=nvlist_add_boolean.9f
981 link path=usr/share/man/man9f/nvlist_add_byte.9f target=nvlist_add_boolean.9f
982 link path=usr/share/man/man9f/nvlist_add_byte_array.9f \
983     target=nvlist_add_boolean.9f
984 link path=usr/share/man/man9f/nvlist_add_int16.9f target=nvlist_add_boolean.9f
985 link path=usr/share/man/man9f/nvlist_add_int16_array.9f \
986     target=nvlist_add_boolean.9f
987 link path=usr/share/man/man9f/nvlist_add_int32.9f target=nvlist_add_boolean.9f
988 link path=usr/share/man/man9f/nvlist_add_int32_array.9f \
989     target=nvlist_add_boolean.9f
990 link path=usr/share/man/man9f/nvlist_add_int64.9f target=nvlist_add_boolean.9f
991 link path=usr/share/man/man9f/nvlist_add_int64_array.9f \
992     target=nvlist_add_boolean.9f
993 link path=usr/share/man/man9f/nvlist_add_int8.9f target=nvlist_add_boolean.9f
994 link path=usr/share/man/man9f/nvlist_add_int8_array.9f \
995     target=nvlist_add_boolean.9f
996 link path=usr/share/man/man9f/nvlist_add_nvlist.9f \
997     target=nvlist_add_boolean.9f
998 link path=usr/share/man/man9f/nvlist_add_nvlist_array.9f \
999     target=nvlist_add_boolean.9f
1000 link path=usr/share/man/man9f/nvlist_add_nvpair.9f \
1001     target=nvlist_add_boolean.9f
1002 link path=usr/share/man/man9f/nvlist_add_string.9f \
1003     target=nvlist_add_boolean.9f
1004 link path=usr/share/man/man9f/nvlist_add_string_array.9f \
1005     target=nvlist_add_boolean.9f
1006 link path=usr/share/man/man9f/nvlist_add_uint16.9f \
1007     target=nvlist_add_boolean.9f
1008 link path=usr/share/man/man9f/nvlist_add_uint16_array.9f \
1009     target=nvlist_add_boolean.9f
1010 link path=usr/share/man/man9f/nvlist_add_uint32.9f \
1011     target=nvlist_add_boolean.9f
1012 link path=usr/share/man/man9f/nvlist_add_uint32_array.9f \
1013     target=nvlist_add_boolean.9f
1014 link path=usr/share/man/man9f/nvlist_add_uint64.9f \
1015     target=nvlist_add_boolean.9f
1016 link path=usr/share/man/man9f/nvlist_add_uint64_array.9f \
1017     target=nvlist_add_boolean.9f
1018 link path=usr/share/man/man9f/nvlist_add_uint8.9f target=nvlist_add_boolean.9f
1019 link path=usr/share/man/man9f/nvlist_add_uint8_array.9f \
1020     target=nvlist_add_boolean.9f
1021 link path=usr/share/man/man9f/nvlist_dup.9f target=nvlist_alloc.9f
1022 link path=usr/share/man/man9f/nvlist_exists.9f target=nvlist_lookup_nvpair.9f
1023 link path=usr/share/man/man9f/nvlist_free.9f target=nvlist_alloc.9f
1024 link path=usr/share/man/man9f/nvlist_lookup_boolean_array.9f \
1025     target=nvlist_lookup_boolean.9f
1026 link path=usr/share/man/man9f/nvlist_lookup_boolean_value.9f \
1027     target=nvlist_lookup_boolean.9f
1028 link path=usr/share/man/man9f/nvlist_lookup_byte.9f \
1029     target=nvlist_lookup_boolean.9f
1030 link path=usr/share/man/man9f/nvlist_lookup_byte_array.9f \
1031     target=nvlist_lookup_boolean.9f
1032 link path=usr/share/man/man9f/nvlist_lookup_int16.9f \
1033     target=nvlist_lookup_boolean.9f
1034 link path=usr/share/man/man9f/nvlist_lookup_int16_array.9f \
1035     target=nvlist_lookup_boolean.9f
1036 link path=usr/share/man/man9f/nvlist_lookup_int32.9f \
1037     target=nvlist_lookup_boolean.9f
1038 link path=usr/share/man/man9f/nvlist_lookup_int32_array.9f \
1039     target=nvlist_lookup_boolean.9f
1040 link path=usr/share/man/man9f/nvlist_lookup_int64.9f \
1041     target=nvlist_lookup_boolean.9f
1042 link path=usr/share/man/man9f/nvlist_lookup_int64_array.9f \
1043     target=nvlist_lookup_boolean.9f

```



```

1044 link path=usr/share/man/man9f/nvlist_lookup_int8.9f \
1045     target=nvlist_lookup_boolean.9f
1046 link path=usr/share/man/man9f/nvlist_lookup_int8_array.9f \
1047     target=nvlist_lookup_boolean.9f
1048 link path=usr/share/man/man9f/nvlist_lookup_nvlist.9f \
1049     target=nvlist_lookup_boolean.9f
1050 link path=usr/share/man/man9f/nvlist_lookup_nvlist_array.9f \
1051     target=nvlist_lookup_boolean.9f
1052 link path=usr/share/man/man9f/nvlist_lookup_pairs.9f \
1053     target=nvlist_lookup_boolean.9f
1054 link path=usr/share/man/man9f/nvlist_lookup_string.9f \
1055     target=nvlist_lookup_boolean.9f
1056 link path=usr/share/man/man9f/nvlist_lookup_string_array.9f \
1057     target=nvlist_lookup_boolean.9f
1058 link path=usr/share/man/man9f/nvlist_lookup_uint16.9f \
1059     target=nvlist_lookup_boolean.9f
1060 link path=usr/share/man/man9f/nvlist_lookup_uint16_array.9f \
1061     target=nvlist_lookup_boolean.9f
1062 link path=usr/share/man/man9f/nvlist_lookup_uint32.9f \
1063     target=nvlist_lookup_boolean.9f
1064 link path=usr/share/man/man9f/nvlist_lookup_uint32_array.9f \
1065     target=nvlist_lookup_boolean.9f
1066 link path=usr/share/man/man9f/nvlist_lookup_uint64.9f \
1067     target=nvlist_lookup_boolean.9f
1068 link path=usr/share/man/man9f/nvlist_lookup_uint64_array.9f \
1069     target=nvlist_lookup_boolean.9f
1070 link path=usr/share/man/man9f/nvlist_lookup_uint8.9f \
1071     target=nvlist_lookup_boolean.9f
1072 link path=usr/share/man/man9f/nvlist_lookup_uint8_array.9f \
1073     target=nvlist_lookup_boolean.9f
1074 link path=usr/share/man/man9f/nvlist_merge.9f target=nvlist_alloc.9f
1075 link path=usr/share/man/man9f/nvlist_pack.9f target=nvlist_alloc.9f
1076 link path=usr/share/man/man9f/nvlist_remove_all.9f target=nvlist_remove.9f
1077 link path=usr/share/man/man9f/nvlist_size.9f target=nvlist_alloc.9f
1078 link path=usr/share/man/man9f/nvlist_t.9f target=nvlist_add_boolean.9f
1079 link path=usr/share/man/man9f/nvlist_unpack.9f target=nvlist_alloc.9f
1080 link path=usr/share/man/man9f/nvlist_xalloc.9f target=nvlist_alloc.9f
1081 link path=usr/share/man/man9f/nvlist_xdup.9f target=nvlist_alloc.9f
1082 link path=usr/share/man/man9f/nvlist_xpack.9f target=nvlist_alloc.9f
1083 link path=usr/share/man/man9f/nvlist_xunpack.9f target=nvlist_alloc.9f
1084 link path=usr/share/man/man9f/nvpair_name.9f target=nvlist_next_nvpair.9f
1085 link path=usr/share/man/man9f/nvpair_type.9f target=nvlist_next_nvpair.9f
1086 link path=usr/share/man/man9f/nvpair_value_boolean_array.9f \
1087     target=nvpair_value_byte.9f
1088 link path=usr/share/man/man9f/nvpair_value_byte_array.9f \
1089     target=nvpair_value_byte.9f
1090 link path=usr/share/man/man9f/nvpair_value_int16.9f \
1091     target=nvpair_value_byte.9f
1092 link path=usr/share/man/man9f/nvpair_value_int16_array.9f \
1093     target=nvpair_value_byte.9f
1094 link path=usr/share/man/man9f/nvpair_value_int32.9f \
1095     target=nvpair_value_byte.9f
1096 link path=usr/share/man/man9f/nvpair_value_int32_array.9f \
1097     target=nvpair_value_byte.9f
1098 link path=usr/share/man/man9f/nvpair_value_int64.9f \
1099     target=nvpair_value_byte.9f
1100 link path=usr/share/man/man9f/nvpair_value_int64_array.9f \
1101     target=nvpair_value_byte.9f
1102 link path=usr/share/man/man9f/nvpair_value_int8.9f target=nvpair_value_byte.9f
1103 link path=usr/share/man/man9f/nvpair_value_int8_array.9f \
1104     target=nvpair_value_byte.9f
1105 link path=usr/share/man/man9f/nvpair_value_nvlist.9f \
1106     target=nvpair_value_byte.9f
1107 link path=usr/share/man/man9f/nvpair_value_nvlist_array.9f \
1108     target=nvpair_value_byte.9f
1109 link path=usr/share/man/man9f/nvpair_value_string.9f \

```

```

1110     target=nvpair_value_byte.9f
1111 link path=usr/share/man/man9f/nvpair_value_string_array.9f \
1112     target=nvpair_value_byte.9f
1113 link path=usr/share/man/man9f/nvpair_value_uint16.9f \
1114     target=nvpair_value_byte.9f
1115 link path=usr/share/man/man9f/nvpair_value_uint16_array.9f \
1116     target=nvpair_value_byte.9f
1117 link path=usr/share/man/man9f/nvpair_value_uint32.9f \
1118     target=nvpair_value_byte.9f
1119 link path=usr/share/man/man9f/nvpair_value_uint32_array.9f \
1120     target=nvpair_value_byte.9f
1121 link path=usr/share/man/man9f/nvpair_value_uint64.9f \
1122     target=nvpair_value_byte.9f
1123 link path=usr/share/man/man9f/nvpair_value_uint64_array.9f \
1124     target=nvpair_value_byte.9f
1125 link path=usr/share/man/man9f/nvpair_value_uint8.9f \
1126     target=nvpair_value_byte.9f
1127 link path=usr/share/man/man9f/nvpair_value_uint8_array.9f \
1128     target=nvpair_value_byte.9f
1129 link path=usr/share/man/man9f/otherq.9f target=OTHERQ.9f
1130 link path=usr/share/man/man9f/outl.9f target=outb.9f
1131 link path=usr/share/man/man9f/outw.9f target=outb.9f
1132 link path=usr/share/man/man9f/pci_config_getl6.9f target=pci_config_get8.9f
1133 link path=usr/share/man/man9f/pci_config_get32.9f target=pci_config_get8.9f
1134 link path=usr/share/man/man9f/pci_config_get64.9f target=pci_config_get8.9f
1135 link path=usr/share/man/man9f/pci_config_getb.9f target=pci_config_get8.9f
1136 link path=usr/share/man/man9f/pci_config_getl.9f target=pci_config_get8.9f
1137 link path=usr/share/man/man9f/pci_config_getll.9f target=pci_config_get8.9f
1138 link path=usr/share/man/man9f/pci_config_getw.9f target=pci_config_get8.9f
1139 link path=usr/share/man/man9f/pci_config_put16.9f target=pci_config_get8.9f
1140 link path=usr/share/man/man9f/pci_config_put32.9f target=pci_config_get8.9f
1141 link path=usr/share/man/man9f/pci_config_put64.9f target=pci_config_get8.9f
1142 link path=usr/share/man/man9f/pci_config_put8.9f target=pci_config_get8.9f
1143 link path=usr/share/man/man9f/pci_config_putb.9f target=pci_config_get8.9f
1144 link path=usr/share/man/man9f/pci_config_putl.9f target=pci_config_get8.9f
1145 link path=usr/share/man/man9f/pci_config_putll.9f target=pci_config_get8.9f
1146 link path=usr/share/man/man9f/pci_config_putw.9f target=pci_config_get8.9f
1147 link path=usr/share/man/man9f/pci_config_teardown.9f \
1148     target=pci_config_setup.9f
1149 link path=usr/share/man/man9f/pci_ereport_post.9f target=pci_ereport_setup.9f
1150 link path=usr/share/man/man9f/pci_ereport_teardown.9f \
1151     target=pci_ereport_setup.9f
1152 link path=usr/share/man/man9f/pci_restore_config_regs.9f \
1153     target=pci_save_config_regs.9f
1154 link path=usr/share/man/man9f/pm_idle_component.9f target=pm_busy_component.9f
1155 link path=usr/share/man/man9f/pm_lower_power.9f target=pm_raise_power.9f
1156 link path=usr/share/man/man9f/priv_policy_choice.9f target=priv_policy.9f
1157 link path=usr/share/man/man9f/priv_policy_only.9f target=priv_policy.9f
1158 link path=usr/share/man/man9f/proc_ref.9f target=proc_signal.9f
1159 link path=usr/share/man/man9f/proc_unref.9f target=proc_signal.9f
1160 link path=usr/share/man/man9f/qprocsoff.9f target=qprocson.9f
1161 link path=usr/share/man/man9f/qwait_sig.9f target=qwait.9f
1162 link path=usr/share/man/man9f/rd.9f target=RD.9f
1163 link path=usr/share/man/man9f/repinsb.9f target=inb.9f
1164 link path=usr/share/man/man9f/repinsd.9f target=inb.9f
1165 link path=usr/share/man/man9f/repinsw.9f target=inb.9f
1166 link path=usr/share/man/man9f/repoutsb.9f target=outb.9f
1167 link path=usr/share/man/man9f/repoutsd.9f target=outb.9f
1168 link path=usr/share/man/man9f/repoutsw.9f target=outb.9f
1169 link path=usr/share/man/man9f/rmallocmap_wait.9f target=rmallocmap.9f
1170 link path=usr/share/man/man9f/rmfreemap.9f target=rmallocmap.9f
1171 link path=usr/share/man/man9f/rw_destroy.9f target=rwlock.9f
1172 link path=usr/share/man/man9f/rw_downgrade.9f target=rwlock.9f
1173 link path=usr/share/man/man9f/rw_enter.9f target=rwlock.9f
1174 link path=usr/share/man/man9f/rw_exit.9f target=rwlock.9f
1175 link path=usr/share/man/man9f/rw_init.9f target=rwlock.9f

```

```
1176 link path=usr/share/man/man9f/rw_read_locked.9f target=rwlock.9f
1177 link path=usr/share/man/man9f/rw_tryenter.9f target=rwlock.9f
1178 link path=usr/share/man/man9f/rw_tryupgrade.9f target=rwlock.9f
1179 link path=usr/share/man/man9f/samestr.9f target=SAMESTR.9f
1180 link path=usr/share/man/man9f/scsi_dmafree.9f target=scsi_dmaget.9f
1181 link path=usr/share/man/man9f/scsi_dname.9f target=scsi_cname.9f
1184 link path=usr/share/man/man9f/scsi_hba_attach.9f \
1185     target=scsi_hba_attach_setup.9f
1182 link path=usr/share/man/man9f/scsi_hba_detach.9f \
1183     target=scsi_hba_attach_setup.9f
1184 link path=usr/share/man/man9f/scsi_hba_fini.9f target=scsi_hba_init.9f
1185 link path=usr/share/man/man9f/scsi_hba_pkt_free.9f \
1186     target=scsi_hba_pkt_alloc.9f
1187 link path=usr/share/man/man9f/scsi_hba_tran_free.9f \
1188     target=scsi_hba_tran_alloc.9f
1189 link path=usr/share/man/man9f/scsi_ifsetcap.9f target=scsi_ifgetcap.9f
1190 link path=usr/share/man/man9f/scsi_mname.9f target=scsi_cname.9f
1191 link path=usr/share/man/man9f/scsi_pktfree.9f target=scsi_pktalloc.9f
1192 link path=usr/share/man/man9f/scsi_realloc.9f target=scsi_pktalloc.9f
1193 link path=usr/share/man/man9f/scsi_resfree.9f target=scsi_pktalloc.9f
1194 link path=usr/share/man/man9f/scsi_rname.9f target=scsi_cname.9f
1195 link path=usr/share/man/man9f/scsi_sense_asc.9f target=scsi_sense_key.9f
1196 link path=usr/share/man/man9f/scsi_sense_ascq.9f target=scsi_sense_key.9f
1197 link path=usr/share/man/man9f/scsi_sense_cmdspecific_uint64.9f \
1198     target=scsi_ext_sense_fields.9f
1199 link path=usr/share/man/man9f/scsi_sense_info_uint64.9f \
1200     target=scsi_ext_sense_fields.9f
1201 link path=usr/share/man/man9f/scsi_sname.9f target=scsi_cname.9f
1202 link path=usr/share/man/man9f/scsi_unslave.9f target=scsi_unprobe.9f
1203 link path=usr/share/man/man9f/sema_destroy.9f target=semaphore.9f
1204 link path=usr/share/man/man9f/sema_init.9f target=semaphore.9f
1205 link path=usr/share/man/man9f/sema_p.9f target=semaphore.9f
1206 link path=usr/share/man/man9f/sema_p_sig.9f target=semaphore.9f
1207 link path=usr/share/man/man9f/sema_tryp.9f target=semaphore.9f
1208 link path=usr/share/man/man9f/sema_v.9f target=semaphore.9f
1209 link path=usr/share/man/man9f/strcasecmp.9f target=string.9f
1210 link path=usr/share/man/man9f/strchr.9f target=string.9f
1211 link path=usr/share/man/man9f/strcmp.9f target=string.9f
1212 link path=usr/share/man/man9f/strcpy.9f target=string.9f
1213 link path=usr/share/man/man9f/strdup.9f target=string.9f
1214 link path=usr/share/man/man9f/strfree.9f target=string.9f
1215 link path=usr/share/man/man9f/strlcat.9f target=string.9f
1216 link path=usr/share/man/man9f/strncpy.9f target=string.9f
1217 link path=usr/share/man/man9f/strlen.9f target=string.9f
1218 link path=usr/share/man/man9f/strncasecmp.9f target=string.9f
1219 link path=usr/share/man/man9f/strncat.9f target=string.9f
1220 link path=usr/share/man/man9f/strncmp.9f target=string.9f
1221 link path=usr/share/man/man9f/strncpy.9f target=string.9f
1222 link path=usr/share/man/man9f/strnlen.9f target=string.9f
1223 link path=usr/share/man/man9f/strrchr.9f target=string.9f
1224 link path=usr/share/man/man9f/strspn.9f target=string.9f
1225 link path=usr/share/man/man9f/taskq_suspended.9f target=taskq.9f
1226 link path=usr/share/man/man9f/uconv_ul6tou8.9f target=uconv_ul6tou32.9f
1227 link path=usr/share/man/man9f/uconv_u32tou16.9f target=uconv_ul6tou32.9f
1228 link path=usr/share/man/man9f/uconv_u32tou8.9f target=uconv_ul6tou32.9f
1229 link path=usr/share/man/man9f/uconv_u8tou16.9f target=uconv_ul6tou32.9f
1230 link path=usr/share/man/man9f/uconv_u8tou32.9f target=uconv_ul6tou32.9f
1231 link path=usr/share/man/man9f/unfreezeestr.9f target=freezeestr.9f
1232 link path=usr/share/man/man9f/va_copy.9f target=va_arg.9f
1233 link path=usr/share/man/man9f/va_end.9f target=va_arg.9f
1234 link path=usr/share/man/man9f/va_start.9f target=va_arg.9f
1235 link path=usr/share/man/man9f/vcmn_err.9f target=cmn_err.9f
1236 link path=usr/share/man/man9f/wr.9f target=WR.9f
1237 link path=usr/share/man/man9f/zcmn_err.9f target=cmn_err.9f
```

new/usr/src/pkg/manifests/system-kernel.man9s.inc

1

3073 Sat May 24 17:48:27 2014

new/usr/src/pkg/manifests/system-kernel.man9s.inc

4888 Undocument dma_req(9s)

4884 EOF scsi_hba_attach

4886 EOF ddi_dmae_getlim

4887 EOF ddi_iomin

4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)

4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet
9 # at http://www.illumos.org/license/CDDL.
10 #
11 #
12 #
13 # Copyright 2011, Richard Lowe
14 # Copyright 2012 Nexenta Systems, Inc. All rights reserved.
15 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
16 #
```

```
18 file path=usr/share/man/man9s/Intro.9s
19 file path=usr/share/man/man9s/aio_req.9s
20 file path=usr/share/man/man9s/buf.9s
21 file path=usr/share/man/man9s/cb_ops.9s
22 file path=usr/share/man/man9s/copyreq.9s
23 file path=usr/share/man/man9s/copyresp.9s
24 file path=usr/share/man/man9s/datab.9s
25 file path=usr/share/man/man9s/ddi_device_acc_attr.9s
26 file path=usr/share/man/man9s/ddi_dma_attr.9s
27 file path=usr/share/man/man9s/ddi_dma_cookie.9s
28 file path=usr/share/man/man9s/ddi_dma_lim_sparc.9s
29 file path=usr/share/man/man9s/ddi_dma_lim_x86.9s
30 file path=usr/share/man/man9s/ddi_dma_req.9s
31 file path=usr/share/man/man9s/ddi_dmae_req.9s
32 file path=usr/share/man/man9s/ddi_fm_error.9s
33 file path=usr/share/man/man9s/ddi_idevice_cookie.9s
34 file path=usr/share/man/man9s/dev_ops.9s
35 file path=usr/share/man/man9s/devmap_callback_ctl.9s
36 file path=usr/share/man/man9s/fmodsw.9s
37 file path=usr/share/man/man9s/free_rtn.9s
38 file path=usr/share/man/man9s/gld_mac_info.9s
39 file path=usr/share/man/man9s/gld_stats.9s
40 file path=usr/share/man/man9s/gld_mac_info.9s
41 file path=usr/share/man/man9s/gld_stats.9s
42 file path=usr/share/man/man9s/hook_nic_event.9s
43 file path=usr/share/man/man9s/hook_pkt_event.9s
44 file path=usr/share/man/man9s/hook_t.9s
45 file path=usr/share/man/man9s/iocblk.9s
46 file path=usr/share/man/man9s/iovect.9s
47 file path=usr/share/man/man9s/kstat.9s
48 file path=usr/share/man/man9s/kstat_intr.9s
49 file path=usr/share/man/man9s/kstat_io.9s
50 file path=usr/share/man/man9s/kstat_named.9s
51 file path=usr/share/man/man9s/linkblk.9s
52 file path=usr/share/man/man9s/modldrv.9s
53 file path=usr/share/man/man9s/modlinkage.9s
54 file path=usr/share/man/man9s/modlmisc.9s
55 file path=usr/share/man/man9s/modlstrmod.9s
56 file path=usr/share/man/man9s/module_info.9s
57 file path=usr/share/man/man9s/msgb.9s
58 file path=usr/share/man/man9s/net_inject_t.9s
```

new/usr/src/pkg/manifests/system-kernel.man9s.inc

2

```
54 file path=usr/share/man/man9s/net_instance_t.9s
55 file path=usr/share/man/man9s/qband.9s
56 file path=usr/share/man/man9s/qinit.9s
57 file path=usr/share/man/man9s/queclass.9s
58 file path=usr/share/man/man9s/queue.9s
59 file path=usr/share/man/man9s/scsi_address.9s
60 file path=usr/share/man/man9s/scsi_arq_status.9s
61 file path=usr/share/man/man9s/scsi_asc_key_strings.9s
62 file path=usr/share/man/man9s/scsi_device.9s
63 file path=usr/share/man/man9s/scsi_extended_sense.9s
64 file path=usr/share/man/man9s/scsi_hba_tran.9s
65 file path=usr/share/man/man9s/scsi_inquiry.9s
66 file path=usr/share/man/man9s/scsi_pkt.9s
67 file path=usr/share/man/man9s/scsi_status.9s
68 file path=usr/share/man/man9s/streamtab.9s
69 file path=usr/share/man/man9s/stroptions.9s
70 file path=usr/share/man/man9s/tuple.9s
71 file path=usr/share/man/man9s/uiio.9s
72 link path=usr/share/man/man9s/dblk.9s target=datab.9s
73 link path=usr/share/man/man9s/ddi_dma_lim.9s target=ddi_dma_lim_sparc.9s
74 link path=usr/share/man/man9s/intro.9s target=Intro.9s
75 link path=usr/share/man/man9s/mblk.9s target=msgb.9s
```

```

*****
316125 Sat May 24 17:48:28 2014
new/usr/src/uts/common/io/scsi/impl/scsi_hba.c
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dma_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21
22 /*
23  * Copyright (c) 1994, 2010, Oracle and/or its affiliates. All rights reserved.
24  * Copyright 2014 Garrett D'Amore <garrett@damore.org>
25  * Copyright 2012 Garrett D'Amore <garrett@damore.org>. All rights reserved.
26 */
27 #include <sys/note.h>
28
29 /*
30  * Generic SCSI Host Bus Adapter interface implementation
31  */
32 #include <sys/scsi/scsi.h>
33 #include <sys/scsi/generic/sas.h>
34 #include <sys/file.h>
35 #include <sys/disp.h>                /* for minclsyspri */
36 #include <sys/ddi_impldefs.h>
37 #include <sys/ndi_impldefs.h>
38 #include <sys/sunndi.h>
39 #include <sys/ddi.h>
40 #include <sys/sunmdi.h>
41 #include <sys/mdi_impldefs.h>
42 #include <sys/callb.h>
43 #include <sys/epm.h>
44 #include <sys/damap.h>
45 #include <sys/time.h>
46 #include <sys/sunldi.h>
47 #include <sys/fm/protocol.h>
48
49 extern struct scsi_pkt *scsi_init_cache_pkt(struct scsi_address *,
50      struct scsi_pkt *, struct buf *, int, int, int, int,
51      int (*)(caddr_t), caddr_t);
52 extern void scsi_free_cache_pkt(struct scsi_address *, struct scsi_pkt *);
53 extern void scsi_cache_dmafree(struct scsi_address *, struct scsi_pkt *);
54 extern void scsi_sync_cache_pkt(struct scsi_address *, struct scsi_pkt *);
55 extern int modrootloaded;

```

```

57 /*
58  * Round up all allocations so that we can guarantee
59  * long-long alignment. This is the same alignment
60  * provided by kmem_alloc().
61  */
62 #define ROUNDUP(x)      (((x) + 0x07) & ~0x07)
63
64 /* Magic number to track correct allocations in wrappers */
65 #define PKT_WRAPPER_MAGIC 0xall0ced /* allocated correctly */
66
67 kmutex_t      scsi_flag_nointr_mutex;
68 kcondvar_t    scsi_flag_nointr_cv;
69 kmutex_t      scsi_log_mutex;
70
71 /* asynchronous probe barrier deletion data structures */
72 static kmutex_t scsi_hba_barrier_mutex;
73 static kcondvar_t scsi_hba_barrier_cv;
74 static struct scsi_hba_barrier {
75     struct scsi_hba_barrier *barrier_next;
76     clock_t barrier_endtime;
77     dev_info_t *barrier_probe;
78 } *scsi_hba_barrier_list;
79
80 unchanged portion omitted
81
82
83
84
85
86
87
88
89
90
91
92
93 /* ARGSUSED */
94 int
95 scsi_hba_attach(
96     dev_info_t *self,
97     ddi_dma_lim_t *hba_lim,
98     scsi_hba_tran_t *tran,
99     int flags,
100    void *hba_options)
101 {
102     ddi_dma_attr_t hba_dma_attr;
103
104     bzero(&hba_dma_attr, sizeof (ddi_dma_attr_t));
105     hba_dma_attr.dma_attr_burstsizes = hba_lim->dlim_burstsizes;
106     hba_dma_attr.dma_attr_minxfer = hba_lim->dlim_minxfer;
107
108     return (scsi_hba_attach_setup(self, &hba_dma_attr, tran, flags));
109 }
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

new/usr/src/uts/common/io/scsi/impl/scsi_resource.c

1

```
*****
19077 Sat May 24 17:48:28 2014
new/usr/src/uts/common/io/scsi/impl/scsi_resource.c
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21
22 /*
23  * Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25 */
26
27 #include <sys/scsi/scsi.h>
28 #include <sys/vtrace.h>
29
30
31 #define A_TO_TRAN(ap) ((ap)->a_hba_tran)
32 #define P_TO_TRAN(pkt) ((pkt)->pkt_address.a_hba_tran)
33 #define P_TO_ADDR(pkt) (&((pkt)->pkt_address))
34
35 /*
36  * Callback id
37  */
38 uintptr_t scsi_callback_id = 0;
39
40 extern ddi_dma_attr_t scsi_alloc_attr;
41
42 struct buf *
43 scsi_alloc_consistent_buf(struct scsi_address *ap,
44 struct buf *in_bp, size_t datalen, uint_t bflags,
45 int (*callback)(caddr_t), caddr_t callback_arg)
46 {
47     dev_info_t *pdip;
48     struct buf *bp;
49     int kmflag;
50     size_t rlen;
51
52     TRACE_0(TR_FAC_SCSI_RES, TR_SCSI_ALLOC_CONSISTENT_BUF_START,
53 "scsi_alloc_consistent_buf_start");
54
55     if (!in_bp) {
56         kmflag = (callback == SLEEP_FUNC) ? KM_SLEEP : KM_NOSLEEP;
```

new/usr/src/uts/common/io/scsi/impl/scsi_resource.c

2

```
57         if ((bp = getrbuf(kmflag)) == NULL) {
58             goto no_resource;
59         }
60     } else {
61         bp = in_bp;
62
63         /* we are establishing a new buffer memory association */
64         bp->b_flags &= ~(B_PAGEIO | B_PHYS | B_REMAPPED | B_SHADOW);
65         bp->b_proc = NULL;
66         bp->b_pages = NULL;
67         bp->b_shadow = NULL;
68     }
69
70     /* limit bits that can be set by bflags argument */
71     ASSERT(!(bflags & ~(B_READ | B_WRITE)));
72     bflags &= (B_READ | B_WRITE);
73     bp->b_un.b_addr = 0;
74
75     if (datalen) {
76         pdip = (A_TO_TRAN(ap))->tran_hba_dip;
77
78         /*
79          * use i_ddi_mem_alloc() for now until we have an interface to
80          * allocate memory for DMA which doesn't require a DMA handle.
81          * ddi_iopb_alloc() is obsolete and we want more flexibility in
82          * controlling the DMA address constraints.
83          */
84         while (i_ddi_mem_alloc(pdip, &scsi_alloc_attr, datalen,
85 ((callback == SLEEP_FUNC) ? 1 : 0), 0, NULL,
86 &bp->b_un.b_addr, &rlen, NULL) != DDI_SUCCESS) {
87             if (callback == SLEEP_FUNC) {
88                 delay(drv_usec_to_hz(10000));
89             } else {
90                 if (!in_bp)
91                     freerbuf(bp);
92                 goto no_resource;
93             }
94         }
95         bp->b_flags |= bflags;
96         bp->b_bcount = datalen;
97         bp->b_resid = 0;
98
99         TRACE_0(TR_FAC_SCSI_RES, TR_SCSI_ALLOC_CONSISTENT_BUF_END,
100 "scsi_alloc_consistent_buf_end");
101     }
102     return (bp);
103
104 no_resource:
105     if (callback != NULL_FUNC && callback != SLEEP_FUNC) {
106         ddi_set_callback(callback, callback_arg,
107 &scsi_callback_id);
108     }
109     TRACE_0(TR_FAC_SCSI_RES,
110 TR_SCSI_ALLOC_CONSISTENT_BUF_RETURN1_END,
111 "scsi_alloc_consistent_buf_end (return1)");
112     return (NULL);
113 }
114
115 _____unchanged_portion_omitted_____
```

79551 Sat May 24 17:48:29 2014

new/usr/src/uts/common/io/scsi/impl/scsi_subr.c

4888 Undocument dma_req(9s)

4884 EOF scsi_hba_attach

4886 EOF ddi_dmae_getlim

4887 EOF ddi_iomin

4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)

4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free

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.
347      * memory for DMA which doesn't require a DMA handle. ddi_iopb_alloc()
348      * is obsolete and we want more flexibility in controlling the DMA
349      * address constraints.
350      */
351     if (i_ddi_mem_alloc(pdip, &scsi_alloc_attr, datalen,
352                       ((func == SLEEP_FUNC) ? 1 : 0), 0, NULL, &local.b_un.b_addr, &rlen,
353                       NULL) != DDI_SUCCESS) {
354         return (pkt);
355     }
356     if (readflag)
357         local.b_flags = B_READ;
358     local.b_bcount = datalen;
359     pkt = (*tran->tran_init_pkt) (ap, NULL, &local,
360                                 cdblen, statuslen, 0, PKT_CONSISTENT,
361                                 (func == SLEEP_FUNC) ? SLEEP_FUNC : NULL_FUNC, NULL);
362     if (!pkt) {
363         i_ddi_mem_free(local.b_un.b_addr, NULL);
364         if (func != NULL_FUNC) {
365             ddi_set_callback(func, NULL, &scsi_callback_id);
366         }
367     } else {
368         *datap = local.b_un.b_addr;
369     }
370     return (pkt);
371 }

```

unchanged portion omitted

```

*****
476405 Sat May 24 17:48:29 2014
new/usr/src/uts/common/io/scsi/targets/st.c
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
_____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) == SCSI_PROBE_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_SCSI_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.
1694      * memory for DMA which doesn't require a DMA handle. ddi_iopb_alloc()
1695      * is obsolete and we want more flexibility in controlling the DMA
1696      * address constraints.
1697      */
1698     (void) i_ddi_mem_alloc(devp->sd_dev, &st_alloc_attr,
1699             sizeof (struct seq_mode), ((km_flags == KM_SLEEP) ? 1 : 0), 0,
1700             NULL, (caddr_t *)&un->un_mspl, &rlen, NULL);

1701     (void) i_ddi_mem_alloc(devp->sd_dev, &st_alloc_attr,
1702             sizeof (read_pos_data_t), ((km_flags == KM_SLEEP) ? 1 : 0), 0,
1703             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_sw_r_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_____

```



```

*****
5515 Sat May 24 17:48:30 2014
new/usr/src/uts/common/io/warlock/ddi_dki_comm.inc
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22  * Copyright 2007 Sun Microsystems, Inc.
23  * All rights reserved. Use is subject to license terms.
24 */

26 /*
27  * Copyright 2014 Garrett D'Amore <garrett@damore.org>
28  * Copyright 2012 Garrett D'Amore <garrett@damore.org>. All rights reserved.
29 */

30 /*
31  * ddi_dki_comm.inc - Part of a pseudo-kernel to use when analyzing drivers
32  * with warlock.
33  *
34  * The main idea here is to represent all of the ways that the kernel can
35  * call into the driver, so that warlock has the correct view of the call
36  * graph.
37  *
38  * This file represents the stuff in common between the DDI/DKI spec and
39  * the current implementation. It is included by both ddi_dki_{spec,impl}.c
40  *
41  * This is a SPARC version; some functions (e.g. ddi_dma_nextwin) should
42  * be changed for an x86 version.
43  */

45 #include <sys/note.h>
46 #include <sys/devops.h>
47 #include <sys/ddi.h>
48 #include <sys/sunddi.h>
49 #include <sys/proc.h>

51 _NOTE(DATA_READABLE_WITHOUT_LOCK( dev_ops cb_ops bus_ops ))

53 /*
54  * Now define a dev_ops, a cb_ops, and a bus_ops with 0 for each
55  * entry point, so that warlock doesn't complain that these

```

```

56 * function pointers have no bindings.
57 *      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
58 */
59 struct dev_ops *devops_p, warlock_dev_ops = {
60     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
61 };
  _unchanged_portion_omitted_

171 int
172 ddi_dma_map(
173     dev_info_t *a,
174     dev_info_t *b,
175     struct ddi_dma_req *c,
176     ddi_dma_handle_t *d)
177 {
178     struct bus_ops *ops;
179     (*ops->bus_dma_map)(0, 0, 0, 0);
180 }

182 int
183 ddi_dma_setup(
184     dev_info_t *a,
185     struct ddi_dma_req *b,
186     ddi_dma_handle_t *c)
187 {
188     struct bus_ops *ops;
189     (*ops->bus_dma_map)(0, 0, 0, 0);
190 }

192 int
193 ddi_dma_mctl(dev_info_t *a, dev_info_t *b, ddi_dma_handle_t c,
194     enum ddi_dma_ctlops d, off_t *e, size_t *f, caddr_t *g,
195     uint_t h)
196 {
197     struct bus_ops *ops;
198     (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
199 }

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

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 }

```

```
187 int
188 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
188 ddi_dma_sync(ddi_dma_handle_t h, off_t o, size_t l, uint_t whom)
189 {
190     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;
191     (*ops->bus_dma_ctl)(0, 0, 0, 0, 0, 0, 0, 0);
192 }
_____unchanged_portion_omitted_____
```

new/usr/src/uts/common/os/sunddi.c

1

```
*****
248847 Sat May 24 17:48:30 2014
new/usr/src/uts/common/os/sunddi.c
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */

22 /*
23  * Copyright (c) 1990, 2010, Oracle and/or its affiliates. All rights reserved.
24  * Copyright 2014 Garrett D'Amore <garrett@damore.org>
25  * Copyright 2012 Garrett D'Amore <garrett@damore.org>. All rights reserved.
26 */

27 #include <sys/note.h>
28 #include <sys/types.h>
29 #include <sys/param.h>
30 #include <sys/system.h>
31 #include <sys/buf.h>
32 #include <sys/uio.h>
33 #include <sys/cred.h>
34 #include <sys/poll.h>
35 #include <sys/mman.h>
36 #include <sys/kmem.h>
37 #include <sys/model.h>
38 #include <sys/file.h>
39 #include <sys/proc.h>
40 #include <sys/open.h>
41 #include <sys/user.h>
42 #include <sys/t_lock.h>
43 #include <sys/vm.h>
44 #include <sys/stat.h>
45 #include <vm/hat.h>
46 #include <vm/seg.h>
47 #include <vm/seg_vn.h>
48 #include <vm/seg_dev.h>
49 #include <vm/as.h>
50 #include <sys/cmn_err.h>
51 #include <sys/cpuvar.h>
52 #include <sys/debug.h>
53 #include <sys/autoconf.h>
54 #include <sys/sunddi.h>
55 #include <sys/esunddi.h>
```

new/usr/src/uts/common/os/sunddi.c

2

```
56 #include <sys/sunddi.h>
57 #include <sys/kstat.h>
58 #include <sys/conf.h>
59 #include <sys/ddi_impldefs.h> /* include implementation structure defs */
60 #include <sys/ndi_impldefs.h> /* include prototypes */
61 #include <sys/ddi_periodic.h>
62 #include <sys/hwconf.h>
63 #include <sys/pathname.h>
64 #include <sys/modctl.h>
65 #include <sys/epm.h>
66 #include <sys/devctl.h>
67 #include <sys/callb.h>
68 #include <sys/cladm.h>
69 #include <sys/sysevent.h>
70 #include <sys/dacf_impl.h>
71 #include <sys/ddidevmap.h>
72 #include <sys/bootconf.h>
73 #include <sys/disp.h>
74 #include <sys/atomic.h>
75 #include <sys/promif.h>
76 #include <sys/instance.h>
77 #include <sys/sysevent/eventdefs.h>
78 #include <sys/task.h>
79 #include <sys/project.h>
80 #include <sys/taskq.h>
81 #include <sys/devpolicy.h>
82 #include <sys/ctype.h>
83 #include <net/if.h>
84 #include <sys/rctl.h>
85 #include <sys/zone.h>
86 #include <sys/clock_impl.h>
87 #include <sys/ddi.h>
88 #include <sys/modhash.h>
89 #include <sys/sunldi_impl.h>
90 #include <sys/fs/dv_node.h>
91 #include <sys/fs/snode.h>

93 extern pri_t minclsyspri;

95 extern rctl_hdl_t rc_project_locked_mem;
96 extern rctl_hdl_t rc_zone_locked_mem;

98 #ifdef DEBUG
99 static int sunddi_debug = 0;
100 #endif /* DEBUG */

102 /* ddi_umem_unlock miscellaneous */

104 static void i_ddi_umem_unlock_thread_start(void);

106 static kmutex_t ddi_umem_unlock_mutex; /* unlock list mutex */
107 static kcondvar_t ddi_umem_unlock_cv; /* unlock list block/unblock */
108 static kthread_t *ddi_umem_unlock_thread;
109 /*
110  * The ddi_umem_unlock FIFO list. NULL head pointer indicates empty list.
111  */
112 static struct ddi_umem_cookie *ddi_umem_unlock_head = NULL;
113 static struct ddi_umem_cookie *ddi_umem_unlock_tail = NULL;

115 /*
116  * DDI(Sun) Function and flag definitions:
117  */

119 #if defined(__x86)
120 /*
121  * Used to indicate which entries were chosen from a range.
```

```

122 */
123 char    *chosen_reg = "chosen-reg";
124 #endif

126 /*
127 * Function used to ring system console bell
128 */
129 void (*ddi_console_bell_func)(clock_t duration);

131 /*
132 * Creating register mappings and handling interrupts:
133 */

135 /*
136 * Generic ddi_map: Call parent to fulfill request...
137 */

139 int
140 ddi_map(dev_info_t *dp, ddi_map_req_t *mp, off_t offset,
141        off_t len, caddr_t *addrp)
142 {
143     dev_info_t *pdip;

145     ASSERT(dp);
146     pdip = (dev_info_t *)DEVI(dp)->devi_parent;
147     return ((DEVI(pdip)->devi_ops->devo_bus_ops->bus_map)(pdip,
148     dp, mp, offset, len, addrp));
149 }
    unchanged portion omitted

707 #endif

709 /*
710 * DMA/DVMA setup
711 */

713 #if defined(__sparc)
714 static ddi_dma_lim_t standard_limits = {
715     (uint_t)0,        /* addr_t dlim_addr_lo */
716     (uint_t)-1,     /* addr_t dlim_addr_hi */
717     (uint_t)-1,     /* uint_t dlim_cntr_max */
718     (uint_t)1,      /* uint_t dlim_burstsizes */
719     (uint_t)1,      /* uint_t dlim_minxfer */
720     0,              /* uint_t dlim_dmaspeed */
721 };
722 #elif defined(__x86)
723 static ddi_dma_lim_t standard_limits = {
724     (uint_t)0,        /* addr_t dlim_addr_lo */
725     (uint_t)0xfffff, /* addr_t dlim_addr_hi */
726     (uint_t)0,        /* uint_t dlim_cntr_max */
727     (uint_t)0x00000001, /* uint_t dlim_burstsizes */
728     (uint_t)DMA_UNIT_8, /* uint_t dlim_minxfer */
729     (uint_t)0,        /* uint_t dlim_dmaspeed */
730     (uint_t)0x86<<24+0, /* uint_t dlim_version */
731     (uint_t)0xffff,   /* uint_t dlim_adreg_max */
732     (uint_t)0xffff,   /* uint_t dlim_ctreg_max */
733     (uint_t)512,      /* uint_t dlim_granular */
734     (int)1,           /* int dlim_sgllen */
735     (uint_t)0xffffffff /* uint_t dlim_reqsizes */
736 };
738 #endif

713 #if !defined(__sparc)
714 /*
715 * Request bus_dma_ctl parent to fiddle with a dma request.

```

```

716 *
717 * (The sparc version is in sparc_subr.s)
718 */
719 int
720 ddi_dma_mctl(dev_info_t *dip, dev_info_t *rdip,
721             ddi_dma_handle_t handle, enum ddi_dma_ctlops request,
722             off_t *offp, size_t *lenp, caddr_t *objp, uint_t flags)
723 {
724     int (*fp)();

726     if (dip != ddi_root_node())
727         dip = (dev_info_t *)DEVI(dip)->devi_bus_dma_ctl;
728     fp = DEVI(dip)->devi_ops->devo_bus_ops->bus_dma_ctl;
729     return ((*fp)(dip, rdip, handle, request, offp, lenp, objp, flags));
730 }
    unchanged portion omitted

934 int
935 ddi_iomin(dev_info_t *a, int i, int stream)
936 {
937     int r;

939     /*
940     * Make sure that the initial value is sane
941     */
942     if (i & (i - 1))
943         return (0);
944     if (i == 0)
945         i = (stream) ? 4 : 1;

947     r = ddi_ctlops(a, a,
948                 DDI_CTLOPS_IOMIN, (void *) (uintptr_t)stream, (void *)&i);
949     if (r != DDI_SUCCESS || (i & (i - 1)))
950         return (0);
951     return (i);
952 }

907 /*
908 * Given two DMA attribute structures, apply the attributes
909 * of one to the other, following the rules of attributes
910 * and the wishes of the caller.
911 *
912 * The rules of DMA attribute structures are that you cannot
913 * make things *less* restrictive as you apply one set
914 * of attributes to another.
915 *
916 */
917 void
918 ddi_dma_attr_merge(ddi_dma_attr_t *attr, ddi_dma_attr_t *mod)
919 {
920     attr->dma_attr_addr_lo =
921         MAX(attr->dma_attr_addr_lo, mod->dma_attr_addr_lo);
922     attr->dma_attr_addr_hi =
923         MIN(attr->dma_attr_addr_hi, mod->dma_attr_addr_hi);
924     attr->dma_attr_count_max =
925         MIN(attr->dma_attr_count_max, mod->dma_attr_count_max);
926     attr->dma_attr_align =
927         MAX(attr->dma_attr_align, mod->dma_attr_align);
928     attr->dma_attr_burstsizes =
929         (uint_t)(attr->dma_attr_burstsizes & mod->dma_attr_burstsizes);
930     attr->dma_attr_minxfer =
931         maxbit(attr->dma_attr_minxfer, mod->dma_attr_minxfer);
932     attr->dma_attr_maxxfer =
933         MIN(attr->dma_attr_maxxfer, mod->dma_attr_maxxfer);
934     attr->dma_attr_seg = MIN(attr->dma_attr_seg, mod->dma_attr_seg);
935     attr->dma_attr_sgllen = MIN((uint_t)attr->dma_attr_sgllen,

```

new/usr/src/uts/common/os/sunddi.c

5

```
936         (uint_t)mod->dma_attr_sgllen);
937     attr->dma_attr_granular =
938         MAX(attr->dma_attr_granular, mod->dma_attr_granular);
939 }
_____unchanged_portion_omitted_____
```

```

*****
7185 Sat May 24 17:48:30 2014
new/usr/src/uts/common/sys/ddi_obsolete.h
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 /*
2  * Copyright 2009 Sun Microsystems, Inc. All rights reserved.
3  * Use is subject to license terms.
4  */
5 /*
6  * Copyright 2014 Garrett D'Amore <garrett@damore.org>
7  * Copyright 2012 Garrett D'Amore <garrett@damore.org>. All rights reserved.
8  */
9 #ifndef _SYS_DDI_OBSOLETE_H
10 #define _SYS_DDI_OBSOLETE_H

12 /*
13  * Obsoleted DDI Interfaces
14  */

16 #include <sys/types.h>
17 #include <sys/dditypes.h>
18 #include <sys/sunldi.h>

21 #ifdef __cplusplus
22 extern "C" {
23 #endif

26 /*
27  * The following are intentionally outside of _DDI_STRICT, they are obsolete
28  * and shouldn't be used by drivers, but are still used in the consolidation.
29  * e.g. DDI implementation.
30  */
31 int ddi_iomin(dev_info_t *dip, int initial, int streaming);

26 #ifndef _DDI_STRICT

28 extern long strtol(const char *, char **, int);
29 extern unsigned long strtoul(const char *, char **, int);

39 /* we'd really like to remove this; unbundled nexus drivers might have it */
40 int ddi_dma_map(dev_info_t *dip, dev_info_t *rdip,
41                struct ddi_dma_req *dmareq, ddi_dma_handle_t *handlep);

31 uint8_t ddi_mem_get8(ddi_acc_handle_t handle, uint8_t *host_addr);
32 uint16_t ddi_mem_get16(ddi_acc_handle_t handle, uint16_t *host_addr);
33 uint32_t ddi_mem_get32(ddi_acc_handle_t handle, uint32_t *host_addr);
34 uint64_t ddi_mem_get64(ddi_acc_handle_t handle, uint64_t *host_addr);
35 void ddi_mem_put8(ddi_acc_handle_t handle, uint8_t *dev_addr, uint8_t value);
36 void ddi_mem_put16(ddi_acc_handle_t handle, uint16_t *dev_addr, uint16_t value);
37 void ddi_mem_put32(ddi_acc_handle_t handle, uint32_t *dev_addr, uint32_t value);
38 void ddi_mem_put64(ddi_acc_handle_t handle, uint64_t *dev_addr, uint64_t value);

40 void ddi_mem_rep_get8(ddi_acc_handle_t handle, uint8_t *host_addr,
41                      uint8_t *dev_addr, size_t reccount, uint_t flags);
42 void ddi_mem_rep_get16(ddi_acc_handle_t handle, uint16_t *host_addr,
43                       uint16_t *dev_addr, size_t reccount, uint_t flags);

```

```

44 void ddi_mem_rep_get32(ddi_acc_handle_t handle, uint32_t *host_addr,
45                      uint32_t *dev_addr, size_t reccount, uint_t flags);
46 void ddi_mem_rep_get64(ddi_acc_handle_t handle, uint64_t *host_addr,
47                      uint64_t *dev_addr, size_t reccount, uint_t flags);
48 void ddi_mem_rep_put8(ddi_acc_handle_t handle, uint8_t *host_addr,
49                      uint8_t *dev_addr, size_t reccount, uint_t flags);
50 void ddi_mem_rep_put16(ddi_acc_handle_t handle, uint16_t *host_addr,
51                      uint16_t *dev_addr, size_t reccount, uint_t flags);
52 void ddi_mem_rep_put32(ddi_acc_handle_t handle, uint32_t *host_addr,
53                      uint32_t *dev_addr, size_t reccount, uint_t flags);
54 void ddi_mem_rep_put64(ddi_acc_handle_t handle, uint64_t *host_addr,
55                      uint64_t *dev_addr, size_t reccount, uint_t flags);

57 uint8_t ddi_io_get8(ddi_acc_handle_t handle, uint8_t *dev_addr);
58 uint16_t ddi_io_get16(ddi_acc_handle_t handle, uint16_t *dev_addr);
59 uint32_t ddi_io_get32(ddi_acc_handle_t handle, uint32_t *dev_addr);
60 void ddi_io_put8(ddi_acc_handle_t handle, uint8_t *dev_addr, uint8_t value);
61 void ddi_io_put16(ddi_acc_handle_t handle, uint16_t *dev_addr, uint16_t value);
62 void ddi_io_put32(ddi_acc_handle_t handle, uint32_t *dev_addr, uint32_t value);

64 void ddi_io_rep_get8(ddi_acc_handle_t handle,
65                    uint8_t *host_addr, uint8_t *dev_addr, size_t reccount);
66 void ddi_io_rep_get16(ddi_acc_handle_t handle,
67                    uint16_t *host_addr, uint16_t *dev_addr, size_t reccount);
68 void ddi_io_rep_get32(ddi_acc_handle_t handle,
69                    uint32_t *host_addr, uint32_t *dev_addr, size_t reccount);
70 void ddi_io_rep_put8(ddi_acc_handle_t handle,
71                    uint8_t *host_addr, uint8_t *dev_addr, size_t reccount);
72 void ddi_io_rep_put16(ddi_acc_handle_t handle,
73                    uint16_t *host_addr, uint16_t *dev_addr, size_t reccount);
74 void ddi_io_rep_put32(ddi_acc_handle_t handle,
75                    uint32_t *host_addr, uint32_t *dev_addr, size_t reccount);

77 /* only support older interfaces on 32-bit systems */
78 #ifdef _ILP32
79 #define ddi_mem_getb          ddi_mem_get8
80 #define ddi_mem_getw          ddi_mem_get16
81 #define ddi_mem_getl          ddi_mem_get32
82 #define ddi_mem_getll         ddi_mem_get64
83 #define ddi_mem_rep_getb     ddi_mem_rep_get8
84 #define ddi_mem_rep_getw     ddi_mem_rep_get16
85 #define ddi_mem_rep_getl     ddi_mem_rep_get32
86 #define ddi_mem_rep_getll    ddi_mem_rep_get64
87 #define ddi_mem_putb         ddi_mem_put8
88 #define ddi_mem_putw         ddi_mem_put16
89 #define ddi_mem_putl         ddi_mem_put32
90 #define ddi_mem_putll        ddi_mem_put64
91 #define ddi_mem_rep_putb     ddi_mem_rep_put8
92 #define ddi_mem_rep_putw     ddi_mem_rep_put16
93 #define ddi_mem_rep_putl     ddi_mem_rep_put32
94 #define ddi_mem_rep_putll    ddi_mem_rep_put64
95 #define ddi_io_getb         ddi_io_get8
96 #define ddi_io_getw         ddi_io_get16
97 #define ddi_io_getl         ddi_io_get32
98 #define ddi_io_putb         ddi_io_put8
99 #define ddi_io_putw         ddi_io_put16
100 #define ddi_io_putl         ddi_io_put32
101 #define ddi_getb            ddi_get8
102 #define ddi_getw            ddi_get16
103 #define ddi_getl            ddi_get32
104 #define ddi_getll           ddi_get64
105 #define ddi_rep_getb        ddi_rep_get8
106 #define ddi_rep_getw        ddi_rep_get16
107 #define ddi_rep_getl        ddi_rep_get32
108 #define ddi_rep_getll       ddi_rep_get64
109 #define ddi_putb            ddi_put8

```

```
110 #define ddi_putw          ddi_put16
111 #define ddi_putl          ddi_put32
112 #define ddi_putll        ddi_put64
113 #define ddi_rep_putb     ddi_rep_put8
114 #define ddi_rep_putw    ddi_rep_put16
115 #define ddi_rep_putl    ddi_rep_put32
116 #define ddi_rep_putll   ddi_rep_put64

118 /* These can't be define's since they're not asm routines */
119 void ddi_io_rep_getb(ddi_acc_handle_t handle, uint8_t *host_addr,
120     uint8_t *dev_addr, size_t recount);
121 void ddi_io_rep_getw(ddi_acc_handle_t handle, uint16_t *host_addr,
122     uint16_t *dev_addr, size_t recount);
123 void ddi_io_rep_getl(ddi_acc_handle_t handle, uint32_t *host_addr,
124     uint32_t *dev_addr, size_t recount);
125 void ddi_io_rep_putb(ddi_acc_handle_t handle, uint8_t *host_addr,
126     uint8_t *dev_addr, size_t recount);
127 void ddi_io_rep_putw(ddi_acc_handle_t handle, uint16_t *host_addr,
128     uint16_t *dev_addr, size_t recount);
129 void ddi_io_rep_putl(ddi_acc_handle_t handle, uint32_t *host_addr,
130     uint32_t *dev_addr, size_t recount);

132 int ddi_peekc(dev_info_t *dip, int8_t *addr, int8_t *val_p);
133 int ddi_peeks(dev_info_t *dip, int16_t *addr, int16_t *val_p);
134 int ddi_peekl(dev_info_t *dip, int32_t *addr, int32_t *val_p);
135 int ddi_peekd(dev_info_t *dip, int64_t *addr, int64_t *val_p);
136 int ddi_pokec(dev_info_t *dip, int8_t *addr, int8_t val);
137 int ddi_pokes(dev_info_t *dip, int16_t *addr, int16_t val);
138 int ddi_pokel(dev_info_t *dip, int32_t *addr, int32_t val);
139 int ddi_poked(dev_info_t *dip, int64_t *addr, int64_t val);

141 uint8_t pci_config_getb(ddi_acc_handle_t handle, off_t offset);
142 uint16_t pci_config_getw(ddi_acc_handle_t handle, off_t offset);
143 uint32_t pci_config_getl(ddi_acc_handle_t handle, off_t offset);
144 uint64_t pci_config_getll(ddi_acc_handle_t handle, off_t offset);
145 void pci_config_putb(ddi_acc_handle_t handle, off_t offset, uint8_t value);
146 void pci_config_putw(ddi_acc_handle_t handle, off_t offset, uint16_t value);
147 void pci_config_putl(ddi_acc_handle_t handle, off_t offset, uint32_t value);
148 void pci_config_putll(ddi_acc_handle_t handle, off_t offset, uint64_t value);

150 extern void repinsb(int port, uint8_t *addr, int count);
151 extern void repinsw(int port, uint16_t *addr, int count);
152 extern void repinsd(int port, uint32_t *addr, int count);
153 extern void repoutsb(int port, uint8_t *addr, int count);
154 extern void repoutsw(int port, uint16_t *addr, int count);
155 extern void repoutsd(int port, uint32_t *addr, int count);
156 #endif

158 /* Obsolete LDI event interfaces */
159 extern int ldi_get_eventcookie(ldi_handle_t, char *,
160     ddi_eventcookie_t *);
161 extern int ldi_add_event_handler(ldi_handle_t, ddi_eventcookie_t,
162     void (*handler)(ldi_handle_t, ddi_eventcookie_t, void *, void *),
163     void *, ldi_callback_id_t *);
164 extern int ldi_remove_event_handler(ldi_handle_t, ldi_callback_id_t);

167 #endif /* not _DDI_STRICT */

169 #ifdef __cplusplus
170 }
_____unchanged_portion_omitted_____
```

new/usr/src/uts/common/sys/ddidmareq.h

1

```
*****
25521 Sat May 24 17:48:30 2014
new/usr/src/uts/common/sys/ddidmareq.h
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22 * Copyright (c) 1990, 2010, Oracle and/or its affiliates. All rights reserved.
23 */
24 */
25 * Copyright 2014 Garrett D'Amore <garrett@damore.org>
25 * Copyright 2012 Garrett D'Amore <garrett@damore.org>. All rights reserved.
26 */

28 #ifndef _SYS_DDIDMAREQ_H
29 #define _SYS_DDIDMAREQ_H

31 #ifdef __cplusplus
32 extern "C" {
33 #endif

35 /*
36  * Memory Objects
37  *
38  * Definitions of structures that can describe
39  * an object that can be mapped for DMA.
40  */

42 /*
43  * Structure describing a virtual address
44  */
45 struct v_address {
46     caddr_t      v_addr;        /* base virtual address */
47     struct as    *v_as;        /* pointer to address space */
48     void         *v_priv;      /* priv data for shadow I/O */
49 };
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
```

new/usr/src/uts/common/sys/ddidmareq.h

2

```
509  * defines then it is assumed that one wishes a callback and is providing
510  * a function address.
511  */
512 #ifndef __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
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
697 * system that if it can synchronize the kernel's view faster
698 * than the CPU's view, it can do so, otherwise it acts the
699 * same as DDI_DMA_SYNC_FORCPU. DDI_DMA_SYNC_FORKERNEL might
700 * speed up the synchronization of kernel mappings in case of
701 * non IO-coherent CPU caches.
702 */
703 #define DDI_DMA_SYNC_FORDEV     0x0
704 #define DDI_DMA_SYNC_FORCPU     0x1
705 #define DDI_DMA_SYNC_FORKERNEL  0x2

```

```

707 /*
708  * Bus nexus control functions for DMA
709  */

711 /*
712  * Control operations, defined here so that devops.h can be included
713  * by drivers without having to include a specific SYSDDI implementation
714  * header file.
715  */

717 enum ddi_dma_ctlops {
718     DDI_DMA_FREE,           /* obsolete - do not use      */
719     DDI_DMA_SYNC,          /* obsolete - do not use      */
720     DDI_DMA_HTOC,          /* obsolete - do not use      */
721     DDI_DMA_KVADDR,        /* obsolete - do not use      */
722     DDI_DMA_MOVWIN,        /* obsolete - do not use      */
723     DDI_DMA_REPWIN,        /* obsolete - do not use      */
724     DDI_DMA_GETERR,        /* obsolete - do not use      */
725     DDI_DMA_COFF,          /* obsolete - do not use      */
726     DDI_DMA_NEXTWIN,       /* obsolete - do not use      */
727     DDI_DMA_NEXTSEG,       /* obsolete - do not use      */
728     DDI_DMA_SEGTOC,        /* obsolete - do not use      */
729     DDI_DMA_RESERVE,       /* reserve some DVMA range    */
730     DDI_DMA_RELEASE,       /* free preallocated DVMA range */
731     DDI_DMA_RESETH,        /* obsolete - do not use      */
732     DDI_DMA_CKSYNC,        /* obsolete - do not use      */
733     DDI_DMA_IOPB_ALLOC,    /* obsolete - do not use      */
734     DDI_DMA_IOPB_FREE,     /* obsolete - do not use      */
735     DDI_DMA_SMEM_ALLOC,    /* obsolete - do not use      */
736     DDI_DMA_SMEM_FREE,     /* obsolete - do not use      */
737     DDI_DMA_IOPB_ALLOC,    /* get contiguous DMA-able memory */
738     DDI_DMA_IOPB_FREE,     /* return contiguous DMA-able memory */
739     DDI_DMA_SMEM_ALLOC,    /* get contiguous DMA-able memory */
740     DDI_DMA_SMEM_FREE,     /* return contiguous DMA-able memory */
741     DDI_DMA_SET_SBUS64,    /* 64 bit SBus support        */
742     DDI_DMA_REMAP,         /* remap DVMA buffers after relocation */
743     DDI_DMA_REMAP,         /* remap DMA buffers after relocation */
744 };

745 /*
746  * control ops for DMA engine on motherboard
747  */
748 DDI_DMA_E_ACQUIRE,       /* get channel for exclusive use */
749 DDI_DMA_E_FREE,          /* release channel                */
750 DDI_DMA_E_1STPTY,        /* setup channel for 1st party DMA */
751 DDI_DMA_E_GETCB,         /* get control block for DMA engine */
752 DDI_DMA_E_FREECB,        /* free control blk for DMA engine */
753 DDI_DMA_E_PROG,          /* program channel of DMA engine  */
754 DDI_DMA_E_SWSETUP,       /* setup channel for software control */
755 DDI_DMA_E_SWSTART,       /* software operation of DMA channel */
756 DDI_DMA_E_ENABLE,        /* enable channel of DMA engine    */
757 DDI_DMA_E_STOP,          /* stop a channel of DMA engine    */
758 DDI_DMA_E_DISABLE,       /* disable channel of DMA engine    */
759 DDI_DMA_E_GETCNT,        /* get remaining xfer count        */
760 DDI_DMA_E_GETLIM,        /* obsolete - do not use      */
761 DDI_DMA_E_GETLIM,        /* get DMA engine limits          */
762 DDI_DMA_E_GETATTR        /* get DMA engine attributes       */
763 };

```

unchanged portion omitted

```

*****
      8599 Sat May 24 17:48:31 2014
new/usr/src/uts/common/sys/dma_engine.h
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License, Version 1.0 only
6  * (the "License"). You may not use this file except in compliance
7  * with the License.
8  *
9  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 * or http://www.opensolaris.org/os/licensing.
11 * See the License for the specific language governing permissions
12 * and limitations under the License.
13 *
14 * When distributing Covered Code, include this CDDL HEADER in each
15 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 * If applicable, add the following below this CDDL HEADER, with the
17 * fields enclosed by brackets "[]" replaced with your own identifying
18 * information: Portions Copyright [yyyy] [name of copyright owner]
19 *
20 * CDDL HEADER END
21 */

23 /*
24 * Copyright 2014 Garrett D'Amore <garrett@damore.org>
25 * Copyright 2012 Garrett D'Amore <garrett@damore.org>. All rights reserved.
26 */

27 /*
28 * Copyright 1998 Sun Microsystems, Inc. All rights reserved.
29 * Use is subject to license terms.
30 */

32 /*      Copyright (c) 1990, 1991 UNIX System Laboratories, Inc. */
33 /*      Copyright (c) 1984, 1986, 1987, 1988, 1989, 1990 AT&T */
34 /*      All Rights Reserved */

36 /*      Copyright (c) 1988, 1989 Intel Corp. */
37 /*      All Rights Reserved */

39 #ifndef _SYS_DMAENGINE_H
40 #define _SYS_DMAENGINE_H

42 #include <sys/types.h>
43 #include <sys/ddtypes.h>

45 #ifdef __cplusplus
46 extern "C" {
47 #endif

49 #define NCHANS 8

51 /*
52  * the DMA Engine Request structure
53  */
54 struct ddi_dmae_req {
55     dev_info_t *der_rdp; /* original requester's dev_info_t */

```

```

56     uchar_t der_command; /* Read/Write/Translate/Verify */
57     uchar_t der_bufprocess; /* NoAuto_init/Chain/Auto_init */
58     uchar_t der_step; /* Inc / Dec / Hold */
59     uchar_t der_trans; /* Single/Demand/Block/Cascade */
60     uchar_t der_path; /* 8/16/32 */
61     uchar_t der_cycles; /* 1 or 2 */
62     uchar_t der_dest; /* Memory / IO */
63     uchar_t der_arbus; /* MicroChannel arbitration reg */
64     ushort_t der_ioadr; /* MicroChannel i/o address reg */
65     ddi_dma_cookie_t *(*proc)(); /* address of application call routine */
66     void *procparms; /* parameter buffer for appl call */
67 };
    unchanged_portion_omitted

136 /*
137  * DMA Engine DDI functions
138  */

140 /*
141  * Get DMA engine limits
142  *
143  * The limits of the DMA engine of the parent bus-nexus are copied into the
144  * provided structure. This should be called at driver attach time,
145  * rather than for each dma setup (breakup).
146  */

148 int ddi_dmae_getlim(dev_info_t *dip, ddi_dma_lim_t *limitsp);

150 /*
141  * Get DMA engine attributes
142  *
143  * The attributes of the DMA engine of the parent bus-nexus are copied into
144  * the provided structure. This should be called at driver attach time,
145  * rather than for each DMA bind.
146  */

148 int ddi_dmae_getattr(dev_info_t *dip, ddi_dma_attr_t *attrp);

150 /*
151  * DMA channel allocation
152  *
153  * The allocation function must be called prior to any other DMA engine
154  * function on a channel. The channel should be freed after completion of the
155  * DMA / device operation if the channel is to be shared.
156  *
157  * Specifics of arguments to ddi_dmae_alloc:
158  *
159  * dip - dev_info pointer, which identifies the base device that wishes
160  * to use the DMA channel.
161  *
162  * chnl - a DMA channel number.
163  *
164  * dmae_waitfp - wait/callback_function pointer, which operates in the same
165  * manner as in ddi_dma_setup(). The value DDI_DMA_DONTWAIT will cause an
166  * immediate return if the channel cannot be acquired. The value
167  * DDI_DMA_SLEEP will will cause the thread to sleep and not return until
168  * the channel has been acquired. Any other value is assumed to be a
169  * callback function address.
170  *
171  * When resources might be available, the callback function is called
172  * (with the argument specified in arg) from interrupt context.
173  *
174  * When the callback function dmae_waitfp() is called, it should attempt to
175  * allocate the DMA channel again. If it succeeds or does not need the
176  * channel any more, it must return the value DDI_DMA_CALLBACK_DONE.

```

```

177 * If it does not want to allocate the channel, but instead wishes to be
178 * called back again later, it must return the value DDI_DMA_CALLBACK_LATER.
179 * If it tries to allocate the channel, but fails to do so, it must return the
180 * value DDI_DMA_CALLBACK_RUNOUT.
181 *
182 * Failure to observe this protocol will have unpredictable results.
183 *
184 * The callback function must provide its own data structure integrity
185 * when it is invoked.
186 */

188 int ddi_dmae_alloc(dev_info_t *dip, int chnl, int (*dmae_waitfp)(),
189     caddr_t arg);

191 /*
192 * DMA channel deallocation
193 *
194 * The deallocation function should be called after completion of the
195 * DMA / device operation if the channel is to be shared.
196 */

198 int ddi_dmae_release(dev_info_t *dip, int chnl);

200 /*
201 * DMA channel used in 1st party DMA scheme
202 *
203 * The specified channel will be configured to operate in a "slave" mode
204 * to a first_party DMA engine that also uses the channel.
205 */

207 int ddi_dmae_1stparty(dev_info_t *dip, int chnl);

209 /*
210 * Program DMA channel
211 *
212 * The DMA channel is setup for an operation using ddi_dmae_prog().
213 * This function is implemented to access all capabilities of the DMA engine
214 * hardware. This function disables the channel prior to setup, and enables
215 * the channel before returning.
216 *
217 * Specifics of arguments to ddi_dmae_prog:
218 *
219 * dmaereqp - pointer to a DMA engine request structure. This structure
220 * is implementation specific and contains all the info necessary to
221 * setup the channel, except for the memory address and count.
222 * This structure is implemented with default values equal to zero,
223 * so that normally only der_command has to be set with a read or write
224 * command value. Once the channel has been setup, subsequent calls to
225 * ddi_dmae_prog() can have dmaereqp set to NULL if only the address and
226 * count have to be updated.
227 *
228 * cookiep - pointer to a ddi_dma_cookie object which contains address,
229 * count and intermediate memory mapping information.
230 */

232 int ddi_dmae_prog(dev_info_t *dip, struct ddi_dmae_req *dmaereqp,
233     ddi_dma_cookie_t *cookiep, int chnl);

235 int ddi_dmae_swsetup(dev_info_t *dip, struct ddi_dmae_req *dmaereqp,
236     ddi_dma_cookie_t *cookiep, int chnl);

238 int ddi_dmae_swstart(dev_info_t *dip, int chnl);

240 /*
241 * Stop DMA channel
242 */

```

```

243 * The DMA channel is disabled and any active operation is terminated.
244 */

246 int ddi_dmae_stop(dev_info_t *dip, int chnl);

248 /*
249 * Enable DMA channel
250 *
251 * The DMA channel is enabled for operation. The channel is also enabled
252 * after successful setup in ddi_dmae_prog().
253 */

255 int ddi_dmae_enable(dev_info_t *dip, int chnl);

257 /*
258 * Disable DMA channel
259 *
260 * The DMA channel is disabled so that transfers cannot continue.
261 */

263 int ddi_dmae_disable(dev_info_t *dip, int chnl);

265 /*
266 * Get remaining xfer count
267 *
268 * The count register of the DMA channel is read. The channel is assumed
269 * to be stopped.
270 */

272 int ddi_dmae_getcnt(dev_info_t *dip, int chnl, int *count);

274 #ifdef __cplusplus
275 }

```

unchanged portion omitted

```

*****
9867 Sat May 24 17:48:31 2014
new/usr/src/uts/common/sys/scsi/conf/device.h
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22 * Copyright 2010 Sun Microsystems, Inc. All rights reserved.
23 * Use is subject to license terms.
24 */
25 /*
26 * Copyright 2014 Garrett D'Amore <garrett@damore.org>
27 */

29 /*
30 * SCSI device structure.
31 *
32 * All SCSI target drivers will have one of these per target/lun/sfunc.
33 * It is allocated and initialized by the framework SCSI HBA nexus code
34 * for each SCSI target dev_info_t node during HBA nexus DDI_CTLOPS_INITCHILD
35 * processing of a child device node just prior to tran_tgt_init(9E). A
36 * pointer to the scsi_device(9S) structure is stored in the
37 * driver-private data field of the target device's dev_info_t node (in
38 * 'devi_driver_data') and can be retrieved by ddi_get_driver_private(9F).
39 */
40 #ifndef _SYS_SCSI_CONF_DEVICE_H
41 #define _SYS_SCSI_CONF_DEVICE_H

43 #include <sys/scsi/scsi_types.h>

45 #ifdef __cplusplus
46 extern "C" {
47 #endif

49 struct scsi_device {
50     /*
51      * Routing information for a SCSI device (target/lun/sfunc).
52      *
53      * The scsi_address(9S) structure contains a pointer to the
54      * scsi_hba_tran(9S) of the transport.
55      *
56      * For devices below an HBA that uses SCSI_HBA_ADDR_SPI

```

```

57     * unit-addressing, the scsi_address(9S) information contains
58     * decoded target/lun addressing information.
59     *
60     * For devices below an HBA that uses SCSI_HBA_ADDR_COMPLEX
61     * unit-addressing, the scsi_address(9S) information contains a
62     * pointer to the scsi_device(9S) structure and the HBA can maintain
63     * its private per-unit-address/per-scsi_device information using
64     * scsi_address_device(9F) and scsi_device_hba_private_[gs]et(9F).
65     *
66     * NOTE: The scsi_address(9S) structure gets structure-copied into
67     * the scsi_pkt(9S) 'pkt_address' field. Having a pointer to the
68     * scsi_device(9S) structure within the scsi_address(9S) allows
69     * the SCSI framework to reflect generic changes in device state
70     * at scsi_pkt_comp(9F) time (given just a scsi_pkt(9S) pointer).
71     *
72     * NOTE: The older SCSI_HBA_TRAN_CLONE method of supporting
73     * SCSI-3 devices is still supported, but use is discouraged.
74     */
75     struct scsi_address      sd_address;

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

80     /*
81     * Target driver mutex for this device. Initialized by SCSI HBA
82     * framework code prior to probe(9E) or attach(9E) of scsi_device.
83     */
84     kmutex_t                sd_mutex;

86     /*
87     * SCSI private: use is associated with implementation of
88     * SCSI_HBA_ADDR_COMPLEX scsi_device_hba_private_[gs]et(9F).
89     * The HBA driver can store a pointer to per-scsi_device(9S)
90     * HBA private data during its tran_tgt_init(9E) implementation
91     * by calling scsi_device_hba_private_set(9F), and free that
92     * pointer during tran_tgt_fini(9E). At tran_send(9E) time, the
93     * HBA driver can use scsi_address_device(9F) to obtain a pointer
94     * to the scsi_device(9S) structure, and then gain access to
95     * its per-scsi_device(9S) hba private data by calling
96     * scsi_device_hba_private_get(9F).
97     */
98     void                    *sd_hba_private;

100     /*
101     * If scsi_slave is used to probe out this device, a scsi_inquiry data
102     * structure will be allocated and an INQUIRY command will be run to
103     * fill it in.
104     *
105     * The allocation will be done via ddi_iopb_alloc, so any manual
106     * freeing may be done by ddi_iopb_free.
107     *
108     * The inquiry data is allocated/refreshed by scsi_probe/scsi_slave
109     * and freed by uninitchild (inquiry data is no longer freed by
110     * scsi_unprobe/scsi_unslave).
111     *
112     * NOTE: Additional device identity information may be available
113     * as properties of sd_dev.
114     */
115     struct scsi_inquiry      *sd_inq;

117     /*
118     * Place to point to an extended request sense buffer.
119     * The target driver is responsible for managing this.
120     */
121     struct scsi_extended_sense *sd_sense;

```

```

120  /*
121  * Target driver 'private' information. Typically a pointer to target
122  * driver private ddi_soft_state(9F) information for the device. This
123  * information is typically established in target driver attach(9E),
124  * and freed in the target driver detach(9E).
125  *
126  * LEGACY: For a scsi_device structure allocated by scsi_vhci during
127  * online of a path, this was set by scsi_vhci to point to the
128  * pathinfo node. Please use sd_pathinfo instead.
129  */
130  void                *sd_private;

132  /*
133  * FMA capabilities of scsi_device.
134  */
135  int                 sd_fm_capable;

137  /*
138  * mdi_pathinfo_t pointer to pathinfo node for scsi_device structure
139  * allocated by the scsi_vhci for transport to a specific pHCI path.
140  */
141  void                *sd_pathinfo;

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

157  /*
158  * The 'sd_tran_safe' field is a grotty hack that allows direct-access
159  * (non-scsa) drivers (like chs, ata, and mlx - which all make cmdk
160  * children) to *illegally* put their own vector in the scsi_address(9S)
161  * 'a_hba_tran' field. When all the drivers that overwrite
162  * 'a_hba_tran' are fixed, we can remove sd_tran_safe (and make
163  * scsi_hba.c code trust that the 'sd_address.a_hba_tran' established
164  * during initchild is still valid when uninitchild occurs).
165  *
166  * NOTE: This hack is also shows up in the DEVP_TO_TRAN implementation
167  * in scsi_confsubr.c.
168  *
169  * NOTE: The 'sd_tran_safe' field is only referenced by SCSI framework
170  * code, so always keeping it at the end of the scsi_device structure
171  * (until it can be removed) is OK. It use to be called 'sd_reserved'.
172  */
173  struct scsi_hba_tran *sd_tran_safe;

175 #ifdef SCSI_SIZE_CLEAN_VERIFY
176  /*
177  * Must be last: Building a driver with-and-without
178  * -DSCSI_SIZE_CLEAN_VERIFY, and checking driver modules for
179  * differences with a tools like 'wsdiff' allows a developer to verify
180  * that their driver has no dependencies on scsi*(9S) size.
181  */
182  int                 _pad[8];
183 #endif /* SCSI_SIZE_CLEAN_VERIFY */
184 };

```

unchanged portion omitted

```

*****
17043 Sat May 24 17:48:31 2014
new/usr/src/uts/common/sys/scsi/impl/transport.h
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22  * Copyright (c) 1990, 2010, Oracle and/or its affiliates. All rights reserved.
23  * Copyright 2014 Garrett D'Amore <garrett@damore.org>
24 */

26 #ifndef _SYS_SCSI_IMPL_TRANSPORT_H
27 #define _SYS_SCSI_IMPL_TRANSPORT_H

29 /*
30  * Include the loadable module wrapper.
31  */
32 #include <sys/modctl.h>
33 #include <sys/note.h>

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

39 #ifdef _KERNEL

41 /*
42  * Opaque handles to address maps
43  */
44 typedef struct __scsi_iportmap  scsi_hba_iportmap_t;
45 typedef struct __scsi_tgtmap    scsi_hba_tgtmap_t;

47 /*
48  * SCSI transport structures
49  *
50  * As each Host Adapter makes itself known to the system,
51  * it will create and register with the library the structure
52  * described below. This is so that the library knows how to route
53  * packets, resource control requests, and capability requests
54  * for any particular host adapter. The 'a_hba_tran' field of a
55  * scsi_address structure made known to a Target driver will
56  * point to one of these transport structures.

```

```

57 */
59 typedef struct scsi_hba_tran  scsi_hba_tran_t;

61 struct scsi_hba_tran {
62     /*
63      * Ptr to the device info structure for this particular HBA. If a SCSI
64      * HBA driver separates initiator port function from HBA function,
65      * this field still refers to the HBA and is used to manage DMA.
66      */
67     dev_info_t      *tran_hba_dip;

69     /*
70      * Private fields for use by the HBA itself.
71      */
72     void            *tran_hba_private;    /* HBA softstate */

74     /*
75      * The following two fields are only used in the deprecated
76      * SCSI_HBA_TRAN_CLONE case. Use SCSI_HBA_ADDR_COMPLEX instead.
77      */
78     void            *tran_tgt_private;
79     struct scsi_device *tran_sd;

81     /*
82      * Vectors to point to specific HBA entry points
83      */
84     int             (*tran_tgt_init)(
85                     dev_info_t      *hba_dip,
86                     dev_info_t      *tgt_dip,
87                     scsi_hba_tran_t *tran,
88                     struct scsi_device *sd);

90     int             (*tran_tgt_probe)(
91                     struct scsi_device *sd,
92                     int (*callback)(
93                             void));

94     void            (*tran_tgt_free)(
95                     dev_info_t      *hba_dip,
96                     dev_info_t      *tgt_dip,
97                     scsi_hba_tran_t *tran,
98                     struct scsi_device *sd);

100    int             (*tran_start)(
101                     struct scsi_address *ap,
102                     struct scsi_pkt *pkt);

104    int             (*tran_reset)(
105                     struct scsi_address *ap,
106                     int level);

108    int             (*tran_abort)(
109                     struct scsi_address *ap,
110                     struct scsi_pkt *pkt);

112    int             (*tran_getcap)(
113                     struct scsi_address *ap,
114                     char *cap,
115                     int whom);

117    int             (*tran_setcap)(
118                     struct scsi_address *ap,
119                     char *cap,
120                     int value,
121                     int whom);

```

```

123     struct scsi_pkt *(*tran_init_pkt)(
124         struct scsi_address *ap,
125         struct scsi_pkt *pkt,
126         struct buf *bp,
127         int cmdlen,
128         int statuslen,
129         int tgtlen,
130         int flags,
131         int (*callback)(
132             caddr_t arg),
133         caddr_t callback_arg);

135     void (*tran_destroy_pkt)(
136         struct scsi_address *ap,
137         struct scsi_pkt *pkt);

139     void (*tran_dmafree)(
140         struct scsi_address *ap,
141         struct scsi_pkt *pkt);

143     void (*tran_sync_pkt)(
144         struct scsi_address *ap,
145         struct scsi_pkt *pkt);

147     int (*tran_reset_notify)(
148         struct scsi_address *ap,
149         int flag,
150         void (*callback)(caddr_t),
151         caddr_t arg);

153     int (*tran_get_bus_addr)(
154         struct scsi_device *sd,
155         char *name,
156         int len);

158     int (*tran_get_name)(
159         struct scsi_device *sd,
160         char *name,
161         int len);

163     int (*tran_clear_aca)(
164         struct scsi_address *ap);

166     int (*tran_clear_task_set)(
167         struct scsi_address *ap);

169     int (*tran_terminate_task)(
170         struct scsi_address *ap,
171         struct scsi_pkt *pkt);

173     int (*tran_get_eventcookie)(
174         dev_info_t *hba_dip,
175         dev_info_t *tgt_dip,
176         char *name,
177         ddi_eventcookie_t *eventp);

179     int (*tran_add_eventcall)(
180         dev_info_t *hba_dip,
181         dev_info_t *tgt_dip,
182         ddi_eventcookie_t event,
183         void (*callback)(
184             dev_info_t *tgt_dip,
185             ddi_eventcookie_t event,
186             void *arg,
187             void *bus_impldata),
188         void *arg,

```

```

189         ddi_callback_id_t *cb_id);

191     int (*tran_remove_eventcall)(dev_info_t *devi,
192         ddi_callback_id_t cb_id);

194     int (*tran_post_event)(
195         dev_info_t *hba_dip,
196         dev_info_t *tgt_dip,
197         ddi_eventcookie_t event,
198         void *bus_impldata);

200     int (*tran_quiesce)(
201         dev_info_t *hba_dip);

203     int (*tran_unquiesce)(
204         dev_info_t *hba_dip);

206     int (*tran_bus_reset)(
207         dev_info_t *hba_dip,
208         int level);

210     /*
211     * Implementation-private specifics.
212     * No HBA should refer to any of the fields below.
213     * This information can and will change.
214     */
215     int tran_hba_flags; /* flag options */

217     uint_t tran_obs1;
218     uchar_t tran_obs2;
219     uchar_t tran_obs3;

221     /*
222     * open_lock: protect tran_minor_isopen
223     * open_flag: bit field indicating which minor nodes are open.
224     * 0 = closed, 1 = shared open, all bits 1 = excl open.
225     *
226     * NOTE: Unused if HBA driver implements its own open(9e) entry point.
227     */
228     kmutex_t tran_open_lock;
229     uint64_t tran_open_flag;

231     /*
232     * bus_config vectors - ON Consolidation Private
233     * These interfaces are subject to change.
234     */
235     int (*tran_bus_config)(
236         dev_info_t *hba_dip,
237         uint_t flag,
238         ddi_bus_config_op_t op,
239         void *arg,
240         dev_info_t **tgt_dipp);

242     int (*tran_bus_unconfig)(
243         dev_info_t *hba_dip,
244         uint_t flag,
245         ddi_bus_config_op_t op,
246         void *arg);

248     int (*tran_bus_power)(
249         dev_info_t *dip,
250         void *impl_arg,
251         pm_bus_power_op_t op,
252         void *arg,
253         void *result);

```



```

255 /*
256  * Inter-Connect type of transport as defined in
257  * usr/src/uts/common/sys/scsi/impl/services.h
258  */
259 int          tran_interconnect_type;

261 /* tran_setup_pkt(9E) related scsi_pkt fields */
262 int          (*tran_pkt_constructor)(
263             struct scsi_pkt          *pkt,
264             scsi_hba_tran_t         *tran,
265             int                      kmflag);
266 void         (*tran_pkt_destructor)(
267             struct scsi_pkt          *pkt,
268             scsi_hba_tran_t         *tran);
269 kmem_cache_t *tran_pkt_cache_ptr;
270 uint_t      tran_hba_len;
271 int         (*tran_setup_pkt)(
272             struct scsi_pkt          *pkt,
273             int                      (*callback)(
274                                     caddr_t arg),
275             caddr_t                 callback_arg);
276 void         (*tran_teardown_pkt)(
277             struct scsi_pkt          *pkt);
278 ddi_dma_attr_t tran_dma_attr;

280 void         *tran_extension;

282 /*
283  * An fm_capable HBA driver can set tran_fm_capable prior to
284  * scsi_hba_attach_setup(). If not set, SCSA provides a default
285  * implementation.
286  */
287 int          tran_fm_capable;

289 /*
290  * Ptr to the device info structure for initiator port. If a SCSA HBA
291  * driver separates initiator port function from HBA function, this
292  * field still refers to the initiator port.
293  */
294 dev_info_t   *tran_iport_dip;

296 /*
297  * map of initiator ports below HBA
298  */
299 scsi_hba_iportmap_t *tran_iportmap;

301 /*
302  * map of targets below initiator
303  */
304 scsi_hba_tgtmap_t   *tran_tgtmap;

306 #ifdef SCSI_SIZE_CLEAN_VERIFY
307 /*
308  * Must be last: Building a driver with-and-without
309  * -DSCSI_SIZE_CLEAN_VERIFY, and checking driver modules for
310  * differences with a tools like 'wsdiff' allows a developer to verify
311  * that their driver has no dependencies on scsi*(9S) size.
312  */
313 int          _pad[8];
314 #endif /* SCSI_SIZE_CLEAN_VERIFY */
315 };
316 size_t      scsi_hba_tran_size();          /* private */

318 #ifdef __lock_lint
319 _NOTE(SCHEME_PROTECTS_DATA("stable data",
320     scsi_hba_tran::tran_sd

```

```

321     scsi_hba_tran::tran_hba_dip
322     scsi_hba_tran::tran_hba_flags
323     scsi_hba_tran::tran_open_flag
324     scsi_hba_tran::tran_pkt_cache_ptr))
325 /*
326  * we only modify the dma attributes (like dma_attr_granular) upon
327  * attach and in response to a setcap. It is also up to the target
328  * driver to not have any outstanding I/Os when it is changing the
329  * capabilities of the transport.
330  */
331 _NOTE(SCHEME_PROTECTS_DATA("serialized by target driver", \
332     scsi_hba_tran::tran_dma_attr.dma_attr_granular))
333 #endif

335 /*
336  * Prototypes for SCSI HBA interface functions
337  *
338  * All these functions are public interfaces, with the
339  * exception of:
340  *   interface          called by
341  *   scsi_initialize_hba_interface()  _init() of scsi module
342  *   scsi_uninitialize_hba_interface() _fini() of scsi module
343  */

345 void          scsi_initialize_hba_interface(void);

347 #ifdef NO SCSI FINI YET
348 void          scsi_uninitialize_hba_interface(void);
349 #endif /* NO SCSI FINI YET */

351 int           scsi_hba_init(
352             struct modlinkage      *modlp);

354 void         scsi_hba_fini(
355             struct modlinkage      *modlp);

356 int          scsi_hba_attach(
357             dev_info_t             *hba_dip,
358             ddi_dma_lim_t          *hba_lim,
359             scsi_hba_tran_t        *tran,
360             int                    flags,
361             void                   *hba_options);

357 int          scsi_hba_attach_setup(
358             dev_info_t             *hba_dip,
359             ddi_dma_attr_t         *hba_dma_attr,
360             scsi_hba_tran_t        *tran,
361             int                    flags);

363 int          scsi_hba_detach(
364             dev_info_t             *hba_dip);

366 scsi_hba_tran_t *scsi_hba_tran_alloc(
367             dev_info_t             *hba_dip,
368             int                    flags);

370 int          scsi_tran_ext_alloc(
371             scsi_hba_tran_t        *tran,
372             size_t                 length,
373             int                    flags);

375 void         scsi_tran_ext_free(
376             scsi_hba_tran_t        *tran,
377             size_t                 length);

379 void         scsi_hba_tran_free(

```

```

380          scsi_hba_tran_t      *tran);
382 int      scsi_hba_probe(
383          struct scsi_device    *sd,
384          int                    (*callback)(void));
386 int      scsi_hba_probe_pi(
387          struct scsi_device    *sd,
388          int                    (*callback)(void),
389          int                    pi);
391 int      scsi_hba_ua_get_reportdev(
392          struct scsi_device    *sd,
393          char                   *ba,
394          int                    len);
396 int      scsi_hba_ua_get(
397          struct scsi_device    *sd,
398          char                   *ua,
399          int                    len);
401 char      *scsi_get_device_type_string(
402          char                   *prop_name,
403          dev_info_t            *hba_dip,
404          struct scsi_device    *sd);
406 int      scsi_get_scsi_maxluns(
407          struct scsi_device    *sd);
409 int      scsi_get_scsi_options(
410          struct scsi_device    *sd,
411          default_scsi_options);
413 int      scsi_get_device_type_scsi_options(
414          dev_info_t            *hba_dip,
415          struct scsi_device    *sd,
416          default_scsi_options);
418 struct scsi_pkt *scsi_hba_pkt_alloc(
419          dev_info_t            *hba_dip,
420          struct scsi_address    *ap,
421          int                    cmdlen,
422          int                    statuslen,
423          int                    tgtlen,
424          int                    hbalen,
425          int                    (*callback)(caddr_t),
426          caddr_t                arg);
428 void      scsi_hba_pkt_free(
429          struct scsi_address    *ap,
430          struct scsi_pkt        *pkt);
433 int      scsi_hba_lookup_capstr(
434          char                   *capstr);
436 int      scsi_hba_in_panic(void);
438 int      scsi_hba_open(
439          dev_t                  *devp,
440          int                    flags,
441          int                    otyp,
442          cred_t                 *credp);
444 int      scsi_hba_close(
445          dev_t                  dev,

```

```

446          int                    flag,
447          int                    otyp,
448          cred_t                 *credp);
450 int      scsi_hba_ioctl(
451          dev_t                  dev,
452          int                    cmd,
453          intptr_t               arg,
454          int                    mode,
455          cred_t                 *credp,
456          int                    *rvalp);
458 void      scsi_hba_nodename_compatible_get(
459          struct scsi_inquiry    *inq,
460          char                   *binding_set,
461          int                    dtype_node,
462          char                   *compat0,
463          char                   **nodenamep,
464          char                   ***compatiblep,
465          int                    *ncompatiblep);
467 void      scsi_hba_nodename_compatible_free(
468          char                   *nodename,
469          char                   **compatible);
471 int      scsi_device_prop_update_inqstring(
472          struct scsi_device    *sd,
473          char                   *name,
474          char                   *data,
475          size_t                 len);
477 void      scsi_hba_pkt_comp(
478          struct scsi_pkt        *pkt);
480 int      scsi_device_identity(
481          struct scsi_device    *sd,
482          int                    (*callback)(void));
484 char      *scsi_hba_iport_unit_address(
485          dev_info_t            *dip);
487 int      scsi_hba_iport_register(
488          dev_info_t            *dip,
489          char                   *port);
491 int      scsi_hba_iport_exist(
492          dev_info_t            *dip);
494 dev_info_t *scsi_hba_iport_find(
495          dev_info_t            *pdip,
496          char                   *portnm);
499 /*
500  * Flags for scsi_hba_attach
501  *
502  * SCSI_HBA_ADDR_SPI          The host adapter driver wants the
503  *                            scsi_address(9S) structure to be maintained
504  *                            in legacy SPI 'a_target'/'a_lun' form (default).
505  *
506  * SCSI_HBA_ADDR_COMPLEX     The host adapter has a complex unit-address
507  *                            space, and the HBA driver wants to maintain
508  *                            per-scsi_device(9S) HBA private data using
509  *                            scsi_address_device(9F) and
510  *                            scsi_device_hba_private_[gs]et(9F). The HBA
511  *                            driver must maintain a private representation

```

```

512 *           of the scsi_device(9S) unit-address - typically
513 *           established during tran_tgt_init(9F) based on
514 *           property values.
515 *
516 * SCSI_HBA_TRAN_PHCI           The host adapter is an mpzio/scsi_vhci pHCI.
517 *           The framework should take care of
518 *           mdi_phci_register() stuff.
519 *
520 * SCSI_HBA_HBA                 The host adapter node (associated with a PCI
521 *           function) is just an HBA, all SCSI initiator
522 *           port function is provided by separate 'iport'
523 *           children of the host adapter node. These iport
524 *           children bind to the same driver as the host
525 *           adapter node. Both nodes are managed by the
526 *           same driver. The driver can distinguish context
527 *           by calling scsi_hba_iport_unit_address().
528 *
529 * ::SCSI_HBA_TRAN_CLONE        Deprecated: use SCSI_HBA_ADDR_COMPLEX instead.
530 *           SCSI_HBA_TRAN_CLONE was a KLUDGE to address
531 *           limitations of the scsi_address(9S) structure
532 *           via duplication of scsi_hba_tran(9S) and
533 *           use of tran_tgt_private.
534 *
535 */
536 #define SCSI_HBA_TRAN_CLONE    0x01    /* Deprecated */
537 #define SCSI_HBA_TRAN_PHCI     0x02    /* treat HBA as mpzio 'pHCI' */
538 #define SCSI_HBA_TRAN_CDB     0x04    /* allocate cdb */
539 #define SCSI_HBA_TRAN_SCB     0x08    /* allocate sense */
540 #define SCSI_HBA_HBA          0x10    /* all HBA children are iports */

542 #define SCSI_HBA_ADDR_SPI      0x20    /* scsi_address in SPI form */
543 #define SCSI_HBA_ADDR_COMPLEX  0x40    /* scsi_address is COMPLEX */

545 /* upper bits used to record SCSI configuration state */
546 #define SCSI_HBA_SCSA_PHCI     0x10000 /* need mdi_phci_unregister */
547 #define SCSI_HBA_SCSA_TA       0x20000 /* scsi_hba_tran_alloc used */
548 #define SCSI_HBA_SCSA_FM       0x40000 /* using common ddi_fm */

550 /*
551 * Flags for scsi_hba allocation functions
552 */
553 #define SCSI_HBA_CANSLEEP      0x01    /* can sleep */

555 /*
556 * Support extra flavors for SCSI children
557 */
558 #define SCSI_FLAVOR_SCSI_DEVICE NDI_FLAVOR_VANILLA
559 #define SCSI_FLAVOR_SMP        1
560 #define SCSI_FLAVOR_IPORT      2
561 #define SCSI_FLAVOR_NFLAVORS   3

563 /*
564 * Maximum number of iport nodes under PCI function
565 */
566 #define SCSI_HBA_MAX_IPORTS    32

568 /*
569 * SCSI iport map interfaces
570 */
571 int     scsi_hba_iportmap_create(
572         dev_info_t           *hba_dip,
573         int                  csync_usec,
574         int                  stable_usec,
575         scsi_hba_iportmap_t **iportmapp);

577 int     scsi_hba_iportmap_iport_add(

```

```

578         scsi_hba_iportmap_t *iportmap,
579         char                  *iport_addr,
580         void                  *iport_priv);

582 int     scsi_hba_iportmap_iport_remove(
583         scsi_hba_iportmap_t *iportmap,
584         char                  *iport_addr);

586 void     scsi_hba_iportmap_destroy(scsi_hba_iportmap_t *iportmap);

588 /*
589 * SCSI target map interfaces
590 */
591 typedef enum {
592     SCSI_TM_FULLSET = 0,
593     SCSI_TM_PERADDR
594 } scsi_tgtmap_mode_t;
595 unchanged portion omitted

```

```

*****
37020 Sat May 24 17:48:31 2014
new/usr/src/uts/i86pc/io/isa.c
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22 * Copyright 2014 Garrett D'Amore <garrett@damore.org>
23 * Copyright (c) 2012 Gary Mills
24 * Copyright (c) 1992, 2010, Oracle and/or its affiliates. All rights reserved.
25 * Copyright 2012 Garrett D'Amore <garrett@damore.org>. All rights reserved.
26 */
27 /*
28 *      ISA bus nexus driver
29 */
30
31 #include <sys/types.h>
32 #include <sys/cmn_err.h>
33 #include <sys/conf.h>
34 #include <sys/modctl.h>
35 #include <sys/autoconf.h>
36 #include <sys/errno.h>
37 #include <sys/debug.h>
38 #include <sys/kmem.h>
39 #include <sys/psm.h>
40 #include <sys/ddidmareq.h>
41 #include <sys/ddi_impldefs.h>
42 #include <sys/ddi_subrdefs.h>
43 #include <sys/dma_engine.h>
44 #include <sys/ddi.h>
45 #include <sys/sunddi.h>
46 #include <sys/sunndi.h>
47 #include <sys/acpi/acpi_enum.h>
48 #include <sys/mach_intr.h>
49 #include <sys/pci.h>
50 #include <sys/note.h>
51 #include <sys/boot_console.h>
52 #include <sys/apic.h>
53 #if defined(__xpv)
54 #include <sys/hypervisor.h>
55 #include <sys/evtchn_impl.h>

```

```

57 extern int console_hypervisor_dev_type(int *);
58 #endif

61 extern int pseudo_isa;
62 extern int isa_resource_setup(void);
63 extern int (*psm_intr_ops)(dev_info_t *, ddi_intr_handle_impl_t *,
64     psm_intr_op_t, int *);
65 extern void pci_register_isa_resources(int, uint32_t, uint32_t);
66 static void isa_enumerate(int);
67 static void enumerate_BIOS_serial(dev_info_t *);
68 static void adjust_prtsz(dev_info_t *isa_dip);
69 static void isa_create_ranges_prop(dev_info_t *);

71 #define USED_RESOURCES "used-resources"

73 /*
74  * The following typedef is used to represent an entry in the "ranges"
75  * property of a pci-isa bridge device node.
76  */
77 typedef struct {
78     uint32_t child_high;
79     uint32_t child_low;
80     uint32_t parent_high;
81     uint32_t parent_mid;
82     uint32_t parent_low;
83     uint32_t size;
84 } pib_ranges_t;
85
86 unchanged portion omitted

121 /* Serial port interrupt vectors for COM1 to COM4. */
122 static int asy_intrs[] = {0x4, 0x3, 0x4, 0x3};
123 /* Bitfield indicating which interrupts are overridden by eeprom config */
124 static uchar_t asy_intr_override = 0;

126 /*
127  *      Local data
128  */
129 static ddi_dma_lim_t ISA_dma_limits = {
130     0, /* address low */
131     0x00ffffff, /* address high */
132     0, /* counter max */
133     1, /* burstsize */
134     DMA_UNIT_8, /* minimum xfer */
135     0, /* dma speed */
136     (uint_t)DMALIM_VERO, /* version */
137     0x0000ffff, /* address register */
138     0x0000ffff, /* counter register */
139     1, /* sector size */
140     0x00000001, /* scatter/gather list length */
141     (uint_t)0xffffffff /* request size */
142 };

130 static ddi_dma_attr_t ISA_dma_attr = {
131     DMA_ATTR_V0,
132     (unsigned long long)0,
133     (unsigned long long)0x00ffffff,
134     0x0000ffff,
135     1,
136     1,
137     1,
138     (unsigned long long)0xffffffff,
139     (unsigned long long)0x0000ffff,
140     1,
141     1,

```

```

142     0
143 };
    unchanged_portion_omitted

561 static int
562 isa_dma_mctl(dev_info_t *dip, dev_info_t *rdip,
563     ddi_dma_handle_t handle, enum ddi_dma_ctlops request,
564     off_t *offp, size_t *lenp, caddr_t *objp, uint_t flags)
565 {
566     int rval;
567     ddi_dma_lim_t default;
568     int arg = (int)(uintptr_t)objp;

569     switch (request) {

571     case DDI_DMA_E_PROG:
572         return (i_dmae_prog(rdip, (struct ddi_dmae_req *)offp,
573             (ddi_dma_cookie_t *)lenp, arg));

575     case DDI_DMA_E_ACQUIRE:
576         return (i_dmae_acquire(rdip, arg, (int*)(caddr_t)offp,
577             (caddr_t)lenp));

579     case DDI_DMA_E_FREE:
580         return (i_dmae_free(rdip, arg));

582     case DDI_DMA_E_STOP:
583         i_dmae_stop(rdip, arg);
584         return (DDI_SUCCESS);

586     case DDI_DMA_E_ENABLE:
587         i_dmae_enable(rdip, arg);
588         return (DDI_SUCCESS);

590     case DDI_DMA_E_DISABLE:
591         i_dmae_disable(rdip, arg);
592         return (DDI_SUCCESS);

594     case DDI_DMA_E_GETCNT:
595         i_dmae_get_chan_stat(rdip, arg, NULL, (int *)lenp);
596         return (DDI_SUCCESS);

598     case DDI_DMA_E_SWSETUP:
599         return (i_dmae_swsetup(rdip, (struct ddi_dmae_req *)offp,
600             (ddi_dma_cookie_t *)lenp, arg));

602     case DDI_DMA_E_SWSTART:
603         i_dmae_swstart(rdip, arg);
604         return (DDI_SUCCESS);

621     case DDI_DMA_E_GETLIM:
622         bcopy(&ISA_dma_limits, objp, sizeof (ddi_dma_lim_t));
623         return (DDI_SUCCESS);

606     case DDI_DMA_E_GETATTR:
607         bcopy(&ISA_dma_attr, objp, sizeof (ddi_dma_attr_t));
608         return (DDI_SUCCESS);

610     case DDI_DMA_E_ISTPTY:
611     {
612         struct ddi_dmae_req reqlstpty =
613             { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
614         if (arg == 0) {
615             reqlstpty.der_command = DMAE_CMD_TRAN;
616             reqlstpty.der_trans = DMAE_TRANS_DMND;
617         } else {

```

```

618         reqlstpty.der_trans = DMAE_TRANS_CSCD;
619     }
620     return (i_dmae_prog(rdip, &reqlstpty, NULL, arg));
621 }

642     case DDI_DMA_IOPB_ALLOC: /* get contiguous DMA-able memory */
643     case DDI_DMA_SMEM_ALLOC:
644         if (!offp) {
645             default = ISA_dma_limits;
646             offp = (off_t *)&default;
647         }
648         /*FALLTHROUGH*/
623     default:
624         /*
625          * We pass to rootnex, but it turns out that rootnex will just
626          * return failure, as we don't use ddi_dma_mctl() except
627          * for DMA engine (ISA) and DVMA (SPARC). Arguably we could
628          * just return an error directly here, instead.
629          */
630         rval = ddi_dma_mctl(dip, rdip, handle, request, offp,
631             lenp, objp, flags);
632     }
633     return (rval);
634 }
    unchanged_portion_omitted

```

new/usr/src/uts/intel/ia32/os/ddi_i86.c

1

45284 Sat May 24 17:48:31 2014

new/usr/src/uts/intel/ia32/os/ddi_i86.c

4888 Undocument dma_req(9s)

4884 EOF scsi_hba_attach

4886 EOF ddi_dmae_getlim

4887 EOF ddi_iomin

4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)

4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free

```
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21
22 /*
23  * Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25 */
26 /*
27  * Copyright 2014 Garrett D'Amore <garrett@damore.org>
28 */
29
30 #include <sys/conf.h>
31 #include <sys/kmem.h>
32 #include <sys/ddi_impldefs.h>
33 #include <sys/ddi.h>
34 #include <sys/sunddi.h>
35 #include <sys/ddifm.h>
36 #include <sys/fm/io/ddi.h>
37 #include <sys/fm/protocol.h>
38 #include <sys/ontrap.h>
39
40
41 /*
42  * DDI DMA Engine functions for x86.
43  * These functions are more naturally generic, but do not apply to SPARC.
44  */
45
46 int
47 ddi_dmae_alloc(dev_info_t *dip, int chnl, int (*dmae_waitfp)(), caddr_t arg)
48 {
49     return (ddi_dma_mctl(dip, dip, 0, DDI_DMA_E_ACQUIRE,
50         (off_t *)dmae_waitfp, (size_t *)arg,
51         (caddr_t *)(&uintptr_t)chnl, 0));
52 }
53
54 unchanged portion omitted
55
56 int
57 ddi_dmae_getlim(dev_info_t *dip, ddi_dma_lim_t *limitsp)
```

new/usr/src/uts/intel/ia32/os/ddi_i86.c

2

```
60 {
61     return (ddi_dma_mctl(dip, dip, 0, DDI_DMA_E_GETLIM, 0, 0,
62         (caddr_t *)limitsp, 0));
63 }
64
65 int
66 ddi_dmae_getattr(dev_info_t *dip, ddi_dma_attr_t *attrp)
67 {
68     return (ddi_dma_mctl(dip, dip, 0, DDI_DMA_E_GETATTR, 0, 0,
69         (caddr_t *)attrp, 0));
70 }
71
72 unchanged portion omitted
```

new/usr/src/uts/intel/io/dktp/dcdev/dadk.c

1

44929 Sat May 24 17:48:32 2014

new/usr/src/uts/intel/io/dktp/dcdev/dadk.c

4888 Undocument dma_req(9s)

4884 EOF scsi_hba_attach

4886 EOF ddi_dmae_getlim

4887 EOF ddi_iomin

4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)

4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free

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.
1054      * memory for DMA which doesn't require a DMA handle. ddi_iopb_alloc()
1055      * is obsolete and we want more flexibility in controlling the DMA
1056      * address constraints..
1057      */
```

```
1058     if (i_ddi_mem_alloc((dadkp->dad_sd)->sd_dev, &dadk_alloc_attr,
1059         (size_t)iobp->b_pbytecnt, ((kmsflg == KM_SLEEP) ? 1 : 0), 0, NULL,
1060         &bp->b_un.b_addr, &rlen, NULL) != DDI_SUCCESS) {
1061         freerbuf(bp);
1062         kmem_free(iobp, sizeof (*iobp));
1063         return (NULL);
1064     }
```

```
1065     iobp->b_flag |= IOB_BPALLOC | IOB_BPBUFALLOC;
1066     iobp->b_bp = bp;
1067     iobp->b_lblk = blkno;
1068     iobp->b_xfer = xfer;
1069     iobp->b_lblk = blkno;
1070     iobp->b_xfer = xfer;
1071     return (iobp);
1072 }
```

unchanged portion omitted

```

*****
49450 Sat May 24 17:48:32 2014
new/usr/src/uts/sun4/os/ddi_impl.c
4888 Undocument dma_req(9s)
4884 EOF scsi_hba_attach
4886 EOF ddi_dmae_getlim
4887 EOF ddi_iomin
4634 undocument scsi_hba_attach() and ddi_dma_lim(9s)
4630 clean stale references to ddi_iopb_alloc and ddi_iopb_free
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */

22 /*
23  * Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25 */
26 /*
27  * Copyright 2014 Garrett D'Amore <garrett@damore.org>
28  * Copyright 2012 Garrett D'Amore <garrett@damore.org>. All rights reserved.
29 */

30 /*
31  * sun4 specific DDI implementation
32 */
33 #include <sys/cpuvar.h>
34 #include <sys/ddi_subrdefs.h>
35 #include <sys/machsystem.h>
36 #include <sys/sunndi.h>
37 #include <sys/sysmacros.h>
38 #include <sys/onttrap.h>
39 #include <vm/seg_kmem.h>
40 #include <sys/membar.h>
41 #include <sys/dditypes.h>
42 #include <sys/ndifm.h>
43 #include <sys/fm/io/ddi.h>
44 #include <sys/ivintr.h>
45 #include <sys/bootconf.h>
46 #include <sys/conf.h>
47 #include <sys/ethernet.h>
48 #include <sys/idprom.h>
49 #include <sys/promif.h>
50 #include <sys/prom_plat.h>
51 #include <sys/systeminfo.h>
52 #include <sys/fpu/fpusystem.h>
53 #include <sys/vm.h>
54 #include <sys/ddi_isa.h>
55 #include <sys/modctl.h>

```

```

57 dev_info_t *get_intr_parent(dev_info_t *, dev_info_t *,
58     ddi_intr_handle_impl_t *);
59 #pragma weak get_intr_parent

61 int process_intr_ops(dev_info_t *, dev_info_t *, ddi_intr_op_t,
62     ddi_intr_handle_impl_t *, void *);
63 #pragma weak process_intr_ops

65 void cells_1275_copy(prop_1275_cell_t *, prop_1275_cell_t *, int32_t);
66 prop_1275_cell_t *cells_1275_cmp(prop_1275_cell_t *, prop_1275_cell_t *,
67     int32_t len);
68 #pragma weak cells_1275_copy

70 /*
71  * Wrapper for ddi_prop_lookup_int_array().
72  * This is handy because it returns the prop length in
73  * bytes which is what most of the callers require.
74 */

76 static int
77 get_prop_int_array(dev_info_t *di, char *pname, int **pval, uint_t *plen)
78 {
79     int ret;

81     if ((ret = ddi_prop_lookup_int_array(DDI_DEV_T_ANY, di,
82         DDI_PROP_DONTPASS, pname, pval, plen)) != DDI_PROP_SUCCESS) {
83         *plen = (*plen) * (uint_t)sizeof (int);
84     }
85     return (ret);
86 }

unchanged_portion_omitted

1134 /*
1135  * This used to be ddi_iomin, but we were the only remaining caller, so
1136  * we've made it private and moved it here.
1137 */
1138 static int
1139 i_ddi_iomin(dev_info_t *a, int i, int stream)
1140 {
1141     int r;

1143     /*
1144      * Make sure that the initial value is sane
1145      */
1146     if (i & (i - 1))
1147         return (0);
1148     if (i == 0)
1149         i = (stream) ? 4 : 1;

1151     r = ddi_ctlops(a, a,
1152         DDI_CTLOPS_IOMIN, (void *)(uintptr_t)stream, (void *)&i);
1153     if (r != DDI_SUCCESS || (i & (i - 1)))
1154         return (0);
1155     return (i);
1156 }

1158 int
1159 i_ddi_mem_alloc(dev_info_t *dip, ddi_dma_attr_t *attr,
1160     size_t length, int cansleep, int flags,
1161     ddi_device_acc_attr_t *accattrp,
1162     caddr_t *kaddrp, size_t *real_length, ddi_acc_hdl_t *handlep)
1163 {
1164     caddr_t a;
1165     int iomin, align, streaming;
1166     uint_t endian_flags = DDI_NEVERSWAP_ACC;

```



```

1168 #if defined(lint)
1169     *handlep = *handlep;
1170 #endif

1172 /*
1173  * Check legality of arguments
1174  */
1175 if (length == 0 || kaddrp == NULL || attr == NULL) {
1176     return (DDI_FAILURE);
1177 }

1179 if (attr->dma_attr_minxfer == 0 || attr->dma_attr_align == 0 ||
1180     (attr->dma_attr_align & (attr->dma_attr_align - 1)) ||
1181     (attr->dma_attr_minxfer & (attr->dma_attr_minxfer - 1))) {
1182     return (DDI_FAILURE);
1183 }

1185 /*
1186  * check if a streaming sequential xfer is requested.
1187  */
1188 streaming = (flags & DDI_DMA_STREAMING) ? 1 : 0;

1190 /*
1191  * Drivers for 64-bit capable SBus devices will encode
1192  * the burtsizes for 64-bit xfers in the upper 16-bits.
1193  * For DMA alignment, we use the most restrictive
1194  * alignment of 32-bit and 64-bit xfers.
1195  */
1196 iomin = (attr->dma_attr_burstsizes & 0xffff) |
1197         ((attr->dma_attr_burstsizes >> 16) & 0xffff);
1198 /*
1199  * If a driver set burtsizes to 0, we give him byte alignment.
1200  * Otherwise align at the burtsizes boundary.
1201  */
1202 if (iomin == 0)
1203     iomin = 1;
1204 else
1205     iomin = 1 << (ddi_fls(iomin) - 1);
1206 iomin = maxbit(iomin, attr->dma_attr_minxfer);
1207 iomin = maxbit(iomin, attr->dma_attr_align);
1208 iomin = i_ddi_iomin(dip, iomin, streaming);
1184 iomin = ddi_iomin(dip, iomin, streaming);
1209 if (iomin == 0)
1210     return (DDI_FAILURE);

1212 ASSERT((iomin & (iomin - 1)) == 0);
1213 ASSERT(iomin >= attr->dma_attr_minxfer);
1214 ASSERT(iomin >= attr->dma_attr_align);

1216 length = P2ROUNDUP(length, iomin);
1217 align = iomin;

1219 if (accattrp != NULL)
1220     endian_flags = accattrp->devacc_attr_endian_flags;

1222 a = kalloca(length, align, cansleep, endian_flags);
1223 if ((*kaddrp = a) == 0) {
1224     return (DDI_FAILURE);
1225 } else {
1226     if (real_length) {
1227         *real_length = length;
1228     }
1229     if (handlep) {
1230         /*
1231         * assign handle information

```

```

1232     */
1233     impl_acc_hdl_init(handlep);
1234 }
1235     return (DDI_SUCCESS);
1236 }
1237 }
unchanged_portion_omitted

```