

MMM		MMM	TTTTTTTTTTTTTTTT		AAAAAAAAA		AAAAAAAAA		CCCCCCCCCCCC		PPPPPPPPPPPP	
MMM		MMM	TTTTTTTTTTTTTTTT		AAAAAAAAA		AAAAAAAAA		CCCCCCCCCCCC		PPPPPPPPPPPP	
MMM		MMM	TTTTTTTTTTTTTTTT		AAAAAAAAA		AAAAAAAAA		CCCCCCCCCCCC		PPPPPPPPPPPP	
MMMMMM	MMMMMM		TTT	AAA	AAA	AAA	AAA	CCC		PPP	PPP	
MMMMMM	MMMMMM		TTT	AAA	AAA	AAA	AAA	CCC		PPP	PPP	
MMMMMM	MMMMMM		TTT	AAA	AAA	AAA	AAA	CCC		PPP	PPP	
MMM	MMM	MMM	TTT	AAA	AAA	AAA	AAA	CCC		PPP	PPP	
MMM	MMM	MMM	TTT	AAA	AAA	AAA	AAA	CCC		PPP	PPP	
MMM	MMM	MMM	TTT	AAA	AAA	AAA	AAA	CCC		PPP	PPP	
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC		PPPPPPPPPPPP		
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC		PPPPPPPPPPPP		
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC		PPPPPPPPPPPP		
MMM		MMM	TTT	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	CCC		PPP		
MMM		MMM	TTT	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	CCC		PPP		
MMM		MMM	TTT	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	CCC		PPP		
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC		PPP		
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC		PPP		
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC		PPP		
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC		PPP		
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCC		PPP		
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCCCCCCCCCCC		PPP		
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCCCCCCCCCCC		PPP		
MMM		MMM	TTT	AAA	AAA	AAA	AAA	CCCCCCCCCCCC		PPP		

CCCCCCCC	NN	NN	TTTTTTTTTT	RRRRRRRR	LL	
CCCCCCCC	NN	NN	TTTTTTTTTT	RRRRRRRR	LL	
CC	NN	NN	TT	RR	RR	LL
CC	NN	NN	TT	RR	RR	LL
CC	NNNN	NN	TT	RR	RR	LL
CC	NNNN	NN	TT	RR	RR	LL
CC	NN	NN	TT	RRRRRRRR		LL
CC	NN	NN	TT	RRRRRRRR		LL
CC	NN	NNNN	TT	RR	RR	LL
CC	NN	NNNN	TT	RR	RR	LL
CC	NN	NN	TT	RR	RR	LL
CC	NN	NN	TT	RR	RR	LL
CC	NN	NN	TT	RR	RR	LL
CC	NN	NN	TT	RR	RR	LL
CCCCCCCC	NN	NN	TT	RR	RR	LLLLLLLLLLLL
CCCCCCCC	NN	NN	TT	RR	RR	LLLLLLLLLLLL

```

LL          IIIIII      SSSSSSSS
LL          IIIIII      SSSSSSSS
LL          II           SS
LL          II           SS
LL          II           SS
LL          II           SS
LL          II           SSSSSS
LL          II           SSSSSS
LL          II           SS
LL          II           SS
LL          II           SS
LL          II           SS
LLLLLLLLLLL IIIIIII    SSSSSSSS
LLLLLLLLLLL IIIIIII    SSSSSSSS

```


L 1
16-Sep-1984 02:11:48
14-Sep-1984 12:46:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MTAACP.SRC]CNTRL.B32;1 Page 1 (1)

CN
VO

```
0001 0
0002 0 MODULE CNTRL (LANGUAGE (BLISS32) ,
0003 0     MAIN = STARTUP
0004 0     IDENT = 'V04-000'
0005 0 ) =
0006 1 BEGIN
0007 1
0008 1 *****
0009 1 *
0010 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 *  ALL RIGHTS RESERVED.
0013 1 *
0014 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 *  TRANSFERRED.
0020 1 *
0021 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 *  CORPORATION.
0024 1 *
0025 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 *****
0030 1
0031 1 ++
0032 1
0033 1 FACILITY: MTAACP
0034 1
0035 1 ABSTRACT:
0036 1     Main control module for MTAACP
0037 1
0038 1 ENVIRONMENT:
0039 1
0040 1     Starlet operating system, including privileged system services
0041 1     and internal exec routines.
0042 1
0043 1 --
0044 1
0045 1
0046 1
0047 1 AUTHOR: D. H. GILLESPIE,      CREATION DATE: 15-MAY-1977
0048 1
0049 1 MODIFIED BY:
0050 1
0051 1     V03-001 MMD0152      Meg Dumont,      26-Apr-1983  9:08
0052 1     Add reference to HDR4 label
0053 1
0054 1     V02-009 DMW00054      David Michael Walp      30-Nov-1981
0055 1     Fixed bug in exception handler which IOSM_REVERSE as
0056 1     IOSM_ACCESS, causing system crash
0057 1
```

CNTRL
V04-000

M 1
16-Sep-1984 02:11:48
14-Sep-1984 12:46:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MTAACP.SRC]CNTRL.B32;1 (1)
Page 2

```

: 58      0058 1  | V02-008 REFORMAT      Maria del C. Nasr      30-Jun-1980
: 59      0059 1  |
: 60      0060 1  | A0007 MCN0003      Maria del C. Nasr      25-Sep-1979  16:32
: 61      0061 1  |      Add HDR3 processing
: 62      0062 1  |
: 63      0063 1  | **
: 64      0064 1  |
: 65      0065 1  | LIBRARY 'SYSS$LIBRARY:LIB.L32';
: 66      0066 1  |
: 67      0067 1  | REQUIRE 'SRC$:MTADEF.B32';
: 68      0451 1  |
: 69      0452 1  | FORWARD ROUTINE
: 70      0453 1  | CANCEL              : COMMON_CALL NOVALUE,      ! signal cancel error
: 71      0454 1  | EXCEPT_HNDLR      : COMMON_CALL NOVALUE,      ! handles exceptions
: 72      0455 1  | MTA_CONTROL         : NOVALUE NOPRES,          ! main control
: 73      0456 1  | RETURN_ALL_ERR      : COMMON_CALL NOVALUE;     ! return all blocked io in error
: 74      0457 1  |
: 75      0458 1  |
```



```
: 77      0459 1 GLOBAL ROUTINE STARTUP : NOVALUE =
: 78      0460 1
: 79      0461 1 ++
: 80      0462 1
: 81      0463 1 FUNCTIONAL DESCRIPTION:
: 82      0464 1     This routine is the startup point for MTAACP. It locks the appropriate
: 83      0465 1     parts of the MTAACP into memory and then calls the control loop in
: 84      0466 1     EXEC mode.
: 85      0467 1
: 86      0468 1 CALLING SEQUENCE:
: 87      0469 1     STARTUP()
: 88      0470 1
: 89      0471 1 INPUT PARAMETERS:
: 90      0472 1     none
: 91      0473 1
: 92      0474 1 IMPLICIT INPUTS:
: 93      0475 1     none
: 94      0476 1
: 95      0477 1 OUTPUT PARAMETERS:
: 96      0478 1     none
: 97      0479 1
: 98      0480 1 IMPLICIT OUTPUTS:
: 99      0481 1     none
: 100     0482 1
: 101     0483 1 ROUTINE VALUE:
: 102     0484 1     none
: 103     0485 1
: 104     0486 1 SIDE EFFECTS:
: 105     0487 1     MTAACP locked into memory, control started
: 106     0488 1
: 107     0489 1 --
: 108     0490 1
: 109     0491 2 BEGIN
: 110     0492 3 EXEC_CALL(MTA_CONTROL)
: 111     0493 1 END;                                     ! end of routine startup
```

```
.TITLE CNTRL
.IDENT \V04-000\
```

```
.EXTRN SYSS$CMEXEC
```

```
.PSECT $CODE$,NOWRT,2
```

```
.ENTRY STARTUP, Save nothing
CLRL -(SP)
PUSHL SP
PUSHAB MTA_CONTROL
CALLS #3, @SYSS$CMEXEC
RET
```

```
: 0459
: 0492
:
:
: 0493
```

```
0000 0000
7E D4 00002
5E DD 00004
CF 9F 00006
03 FB 0000A
04 00011
00000000G 9F 0000V
```

```
; Routine Size: 18 bytes, Routine Base: $CODE$ + 0000
```

```
; 112      0494 1
```

```
114 0495 1 ROUTINE MTA_CONTROL : NOVALUE NOPRES =
115 0496 1
116 0497 1 ++
117 0498 1
118 0499 1 FUNCTIONAL DESCRIPTION:
119 0500 1 Main control module for MTAACP
120 0501 1
121 0502 1 CALLING SEQUENCE:
122 0503 1 MTA_CONTROL()
123 0504 1
124 0505 1 INPUT PARAMETERS:
125 0506 1 none
126 0507 1
127 0508 1 IMPLICIT INPUTS:
128 0509 1 ACP QUEUE
129 0510 1
130 0511 1 OUTPUT PARAMETERS:
131 0512 1 none
132 0513 1
133 0514 1 IMPLICIT OUTPUTS:
134 0515 1 none
135 0516 1
136 0517 1 ROUTINE VALUE:
137 0518 1 none
138 0519 1
139 0520 1 SIDE EFFECTS:
140 0521 1 MTAACP functions executed
141 0522 1
142 0523 1 --
143 0524 1
144 0525 2 BEGIN
145 0526 2
146 0527 2 EXTERNAL REGISTER
147 0528 2 COMMON_REG;
148 0529 2
149 0530 2 GLOBAL
150 0531 2
151 0532 2 ! the following are not initialized for each request
152 0533 2 !
153 0534 2 QUEUE HEAD : REF BBLOCK, ! pter to MTAACP input queue
154 0535 2 DISK_UCB : REF BBLOCK, ! UCB of SYS$DISK
155 0536 2 IO_CHANNEL, ! channel # for I/O
156 0537 2 MAIL_CHANNEL, ! channel # for mailbox
157 0538 2 FREE_PAGE_HEAD : VECTOR [2], ! free page list head
158 0539 2 LAST_PAGE, ! last page of program region in virtual memory
159 0540 2
160 0541 2 ! the following are initialized for each request
161 0542 2 !
162 0543 2 USER_STATUS : VECTOR [2], ! i/o status to user
163 0544 2 IO_STATUS : VECTOR [2], ! status block used by mtaacp
164 0545 2 IO_PACKET : REF BBLOCK, ! addr of i/o request packet
165 0546 2 CURRENT_UCB : REF BBLOCK, ! addr of current ucb
166 0547 2 CURRENT_WCB : REF BBLOCK, ! addr of current wcb
167 0548 2 HDR1 : REF BBLOCK, ! address hdr1 label(eof1)
168 0549 2 HDR2 : REF BBLOCK, ! address hdr2 label(eof2)
169 0550 2 HDR3 : REF BBLOCK, ! address hdr3 label(Eof3)
170 0551 2 HDR4 : REF BBLOCK, ! address hdr4 label(Eof4)
```



```
171 0552 2 LOCAL_FIB : BBLOCK [FIB$K_MTALEN], ! copy of fib
172 0553 2 IMPURE_END : VECTOR [0];
173 0554 2
174 0555 2 GLOBAL_LITERAL
175 0556 2 IMPURE_SIZE = IMPURE_END - USER_STATUS;
176 0557 2
177 0558 2 LOCAL
178 0559 2 FUNCTION; ! function code and qualifiers
179 0560 2
180 0561 2 OWN
181 0562 2 RANGE : VECTOR [2] INITIAL (0, %X'FFFFFFFF');
182 0563 2
183 0564 2 EXTERNAL_ROUTINE
184 0565 2 END_OF_VOL : NOPRES, ! end-of-volume processing
185 0566 2 GET_REQ : LSGET_REQ, ! get next request
186 0567 2 INIT_MTAACP, ! acp initialization
187 0568 2 IO_DONE, ! complete i/o processing
188 0569 2 MTA_ACCESS : NOPRES, ! access a file
189 0570 2 MTA_ACPCTRL : NOPRES, ! mtaacp control functions
190 0571 2 MTA_CREATE : NOPRES, ! create a file
191 0572 2 MTA_DEACCESS : NOPRES, ! deaccess a file
192 0573 2 MTA_MODIFY : NOPRES, ! modify a file
193 0574 2 MTA_MOUNT : NOPRES, ! mount a volume
194 0575 2 SYSSPURGWS : ADDRESSING_MODE (ABSOLUTE);
195 0576 2
196 0577 2 ! enable except_hndlr
197 0578 2
198 0579 2 BEGIN
199 0580 2
200 0581 2 BUILTIN
201 0582 2 FP;
202 0583 2
203 0584 2 .FP = EXCEPT_HNDLR;
204 0585 2 END;
205 0586 2 KERNEL CALL (INIT_MTAACP); ! initialize acp, one time only
206 0587 2 SYSSPURGWS (RANGE);
207 0588 2
208 0589 2 WHILE 1
209 0590 2 DO
210 0591 2 BEGIN
211 0592 2 ! initialize impure area and set user status to success
212 0593 2 !
213 0594 2 CH$FILL (0, IMPURE_SIZE, USER_STATUS);
214 0595 2 USER_STATUS[0] = T;
215 0596 2 IO_PACKET = GET_REQ(); ! get a request
216 0597 2 FUNCTION = .IO_PACKET[IRP$V_FCODE];
217 0598 2
218 0599 2 IF .CURRENT_VCB[VCB$V_CANCELIO] AND .FUNCTION NEQ IOS_DEACCESS
219 0600 2 THEN
220 0601 2 CANCEL()
221 0602 2 ELSE
222 0603 2 IF .FUNCTION EQL IOS_READPBLK OR .FUNCTION EQL IOS_WRITEPBLK
223 0604 2 THEN
224 0605 2 END_OF_VOL() ! end of volume processing
225 0606 2 ELSE
226 0607 2
227 0608 2
```

```

: 228      0609 4      BEGIN
: 229      0610 4
: 230      0611 4      CASE .FUNCTION FROM IOS_ACCESS TO IOS_MOUNT OF
: 231      0612 4      SET
: 232      0613 4
: 233      0614 4      [IOS_ACCESS] :      MTA_ACCESS();
: 234      0615 4
: 235      0616 4      [IOS_CREATE] :      MTA_CREATE();
: 236      0617 4
: 237      0618 4      [IOS_DEACCESS] :      MTA_DEACCESS();
: 238      0619 4
: 239      0620 4      [IOS_DELETE] :      (ERROR(SS$_ILLIOFUNC);0);
: 240      0621 4
: 241      0622 4      [IOS_MODIFY] :      MTA_MODIFY();
: 242      0623 4
: 243      0624 4      [IOS_ACPCONTROL] :      MTA_ACPCTRL();
: 244      0625 4
: 245      0626 4      [IOS_MOUNT] :      MTA_MOUNT();
: 246      0627 4
: 247      0628 4      [INRANGE] :      (ERROR(SS$_ILLIOFUNC);0);
: 248      0629 4
: 249      0630 4      [OUTRANGE] :      (ERROR(SS$_ILLIOFUNC);0);
: 250      0631 4      TES;
: 251      0632 4
: 252      0633 4      END;
: 253      0634 4
: 254      0635 4      IF .IO_PACKET NEQ 0
: 255      0636 4      THEN
: 256      0637 4      KERNEL_CALL(IO_DONE, .IO_PACKET);
: 257      0638 4
: 258      0639 4      END;
: 259      0640 4      ! end of while loop
: 260      0641 1      END;
:                ! end of routine
```

.PSECT \$LOCKEDD1\$,NOEXE,2

```

00000 QUEUE_HEAD::
:                .BLKB 4
00004 DISK_UCB::
:                .BLKB 4
00008 IO_CHANNEL::
:                .BLKB 4
0000C MAIL_CHANNEL::
:                .BLKB 4
00010 FREE_PAGE_HEAD::
:                .BLKB 8
00018 LAST_PAGE::
:                .BLKB 4
0001C USER_STATUS::
:                .BLKB 8
00024 IO_STATUS::
:                .BLKB 8
0002C IO_PACKET::
:                .BLKB 4
00030 CURRENT_UCB::
```



```
00034 CURRENT_WCB:: .BLKB 4
00038 HDR1:: .BLKB 4
0003C HDR2:: .BLKB 4
00040 HDR3:: .BLKB 4
00044 HDR4:: .BLKB 4
00048 LOCAL_FIB:: .BLKB 28
00064 IMPURE_END:: .BLKB 0
FFFFFFFF 00000000 00064 RANGE: .LONG 0, -1
IMPURE_SIZE== 72
.EXTRN END_OF_VOL, GET_REQ
.EXTRN INIT_MTAACP, IO_DONE
.EXTRN MTA_ACCESS, MTA_ACPCNTRL
.EXTRN MTA_CREATE, MTA_DEACCESS
.EXTRN MTA_MODIFY, MTA_MOUNT
.EXTRN SYS$PURGWS, SYS$CMKRNL
.PSECT $CODE$,NOWRT,2
```

```
0000 00000 MTA_CONTROL:
5E 0000V 04 C2 00002 .WORD Save nothing ; 0495
6D 0000V CF 9E 00005 .SUBL2 #4, SP ; 0584
7E D4 0000A .MOVAB EXCEPT_HNDLR, (FP) ; 0586
5E DD 0000C .CLRL -(SP)
0000G CF 9F 0000E .PUSHL SP
0000' CF 9F 00012 .PUSHAB INIT_MTAACP
0000' CF 9F 00019 .CALLS #3, @#SYS$CMKRNL ; 0587
0000' CF 9F 0001D .PUSHAB RANGE
0000' CF 9F 00024 1$: .CALLS #1, @#SYS$PURGWS ; 0595
0000' CF 0002B .MOVCS #0, (SP), #0, #72, USER_STATUS ; 0596
0000G 01 D0 0002E .MOVL #1, USER_STATUS ; 0597
50 D0 00033 .BSBW GET_REQ
00 EF 0003B .MOVL R0, IO_PACKET
05 E1 00041 .EXTZV #0, #6, 32(R0), FUNCTION ; 0598
6E D1 00046 .BBC #5, 11(CURRENT_VCB), 2$ ; 0600
07 13 00049 .CMPL FUNCTION, #52
00 FB 0004B .BEQL 2$
57 11 00050 .CALLS #0, CANCEL ; 0602
6E D1 00052 2$: .BRB 13$ ; 0605
05 13 00055 .CMPL FUNCTION, #12
6E D1 00057 .BEQL 3$
07 12 0005A .CMPL FUNCTION, #11
00 FB 0005C 3$: .BNEQ 4$ ; 0607
46 11 00061 .CALLS #0, END_OF_VOL
6E CF 00063 4$: .BRB 13$ ; 0611
0012 00067 5$: .CASEL FUNCTION, #50, #7
0027 0006F .WORD 6$-5$,-
7$-5$,-
8$-5$,-
12$-5$,-
9$-5$,-
12$-5$,-
```

CNTRL
V04-000

F 2
16-Sep-1984 02:11:48
14-Sep-1984 12:46:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MTAACP.SRC]CNTRL.B32;1
Page 8
(3)

						10\$-5\$,-		
						11\$-5\$		
0000G	CF	2A	11	00077	BRB	12\$		0630
		00	FB	00079 6\$:	CALLS	#0, MTA_ACCESS		0614
		29	11	0007E	BRB	13\$		
0000G	CF	00	FB	00080 7\$:	CALLS	#0, MTA_CREATE		0616
		22	11	00085	BRB	13\$		
0000G	CF	00	FB	00087 8\$:	CALLS	#0, MTA_DEACCESS		0618
		1B	11	0008C	BRB	13\$		
0000G	CF	00	FB	0008E 9\$:	CALLS	#0, MTA_MODIFY		0622
		14	11	00093	BRB	13\$		
0000G	CF	00	FB	00095 10\$:	CALLS	#0, MTA_ACPCNTRL		0624
		0D	11	0009A	BRB	13\$		
0000G	CF	00	FB	0009C 11\$:	CALLS	#0, MTA_MOUNT		0626
		06	11	000A1	BRB	13\$		
0000G	CF	F4	8F	9B 000A3 12\$:	MOVZBW	#244, USER_STATUS		0628
	50	0000'	CF	D0 000A9 13\$:	MOVL	IO PACKET, -R0		0635
			11	13 000AE	BEQL	14\$		
			50	DD 000B0	PUSHL	R0		0637
			01	DD 000B2	PUSHL	#1		
			5E	DD 000B4	PUSHL	SP		
00000000G	9F	0000G	CF	9F 000B6	PUSHAB	IO_DONE		
			04	FB 000BA	CALLS	#4, @#SYSS\$CMKRNL		
		FF60	31	000C1 14\$:	BRW	1\$		0589

; Routine Size: 196 bytes, Routine Base: \$CODE\$ + 0012

; 261 0642 1

CO
VO


```
263 0643 1 GLOBAL ROUTINE EXCEPT_HNDLR (SIGNAL, MECHANISM) : COMMON_CALL NOVALUE =
264 0644 1
265 0645 1 ++
266 0646 1
267 0647 1 FUNCTIONAL DESCRIPTION:
268 0648 1 This routine handles exceptions. This code stores the condition value
269 0649 1 (if given) in the user status block, unwinds the stack, completes
270 0650 1 io and returns.
271 0651 1
272 0652 1
273 0653 1 CALLING SEQUENCE:
274 0654 1 EXCEPT_HNLR(ARG1,ARG2)
275 0655 1
276 0656 1 INPUT PARAMETERS:
277 0657 1 ARG1 - address of signal array
278 0658 1 ARG2 - address of mechanism array
279 0659 1
280 0660 1 IMPLICIT INPUTS:
281 0661 1 none
282 0662 1
283 0663 1 OUTPUT PARAMETERS:
284 0664 1 none
285 0665 1
286 0666 1 IMPLICIT OUTPUTS:
287 0667 1 USER_STATUS - receives exception parameter
288 0668 1
289 0669 1 ROUTINE VALUE:
290 0670 1 none
291 0671 1
292 0672 1 SIDE EFFECTS:
293 0673 1 Stack is unwound to main level to continue processing requests.
294 0674 1
295 0675 1 --
296 0676 1
297 0677 2 BEGIN
298 0678 2
299 0679 2 EXTERNAL REGISTER
300 0680 2 COMMON_REG;
301 0681 2
302 0682 2 LOCAL
303 0683 2 FUNCTION : BLOCK [1]; ! io request function with modifiers
304 0684 2
305 0685 2 MAP
306 0686 2 SIGNAL : REF BBLOCK, ! signal array
307 0687 2 MECHANISM : REF BBLOCK; ! mechanism array
308 0688 2
309 0689 2 EXTERNAL
310 0690 2 CURRENT_WCB : REF BBLOCK, ! address of current window
311 0691 2 IO_PACKET : REF BBLOCK, ! address of current io request packet
312 0692 2 USER_STATUS : WORD; ! status returned to user
313 0693 2
314 0694 2 EXTERNAL ROUTINE
315 0695 2 SYS$UNWIND : ADDRESSING MODE (ABSOLUTE), ! unwind the stack
316 0696 2 TERMINATE_VOL : COMMON_CALL, ! terminate activity on this volume
317 0697 2 ZERO_CHANNEL : COMMON_CALL; ! clean up user channel
318 0698 2
319 0699 2 ! Check the signal code. The only permissible ones are SS$UNWIND which is
```

```

: 320      0700      2      ! ignored and SSS_CMUSER. The error status is the 16 bit CHMU code. If
: 321      0701      2      ! the error value is non_zero, store it in the user status, complete the
: 322      0702      2      ! i/o, unwind the stack and return to the control module. If the error
: 323      0703      2      ! value is zero the user status is not changed but the i/o is completed,
: 324      0704      2      ! the stack is unwound and the return is to the control module.
: 325      0705      2      !
: 326      0706      2      !
: 327      0707      2      IF .SIGNAL[CHFSL_SIG_NAME] EQL SSS_UNWIND
: 328      0708      2      THEN
: 329      0709      2      RETURN;
: 330      0710      2      !
: 331      0711      2      IF .SIGNAL[CHFSL_SIG_NAME] NEQ SSS_CMUSER
: 332      0712      2      THEN
: 333      0713      2      BUG_CHECK(UNXSIGNAL);
: 334      0714      2      !
: 335      0715      2      ! Get IO function and mask out the create bit, then if access modifier bit
: 336      0716      2      ! set on a valid "access type" function THEN clean user channel
: 337      0717      2      !
: 338      0718      2      FUNCTION = .IO_PACKET[IRPSW_FUNC] AND (NOT IOSM_CREATE);
: 339      0719      2      IF (.FUNCTION EQL (IOS_ACCESS OR IOSM_ACCESS))
: 340      0720      2      OR (.FUNCTION EQL (IOS_CREATE OR IOSM_ACCESS))
: 341      0721      2      THEN
: 342      0722      2      KERNEL_CALL(ZERO_CHANNEL, .IO_PACKET);
: 343      0723      2      !
: 344      0724      2      ! error to be returned to user
: 345      0725      2      !
: 346      0726      2      IF .SIGNAL[CHFSL_SIG_ARG1] NEQ 0
: 347      0727      2      THEN
: 348      0728      2      USER_STATUS = .SIGNAL[CHFSL_SIG_ARG1];
: 349      0729      2      !
: 350      0730      2      ! return all blocked io in error
: 351      0731      2      !
: 352      0732      2      IF NOT .CURRENT_VCB[VCBSV_WAIUSRLBL]
: 353      0733      2      THEN
: 354      0734      2      KERNEL_CALL(RETURN_ALL_ERR);
: 355      0735      2      !
: 356      0736      2      ! if tape position lost, then force user to close open files and reposition
: 357      0737      2      ! at beginning of volume set
: 358      0738      2      !
: 359      0739      2      IF .SIGNAL[CHFSL_SIG_ARG1] EQL SSS_TAPEPOSLOST
: 360      0740      2      THEN
: 361      0741      2      KERNEL_CALL(TERMINATE_VOL, .CURRENT_VCB[VCBSL_WCB]);
: 362      0742      2      !
: 363      0743      2      SYSSUNWIND(MECHANISM[CHFSL_MCH_DEPTH], 0);
: 364      0744      2      RETURN;
: 365      0745      2      !
: 366      0746      1      END;
```

```

00000920 53 00000000G 9F 9E 00002
50 04 AC D0 00009
8F 04 A0 D1 0000D
```

```

.EXTRN SYSSUNWIND, TERMINATE_VOL
.EXTRN ZERO_CHANNEL, BUG$UNXSIGNAL

.ENTRY EXCEPT_HNDLR, Save R2,R3
MOVAB @#SYSSCMKRN, R3
MOVL SIGNAL, R0
CMPL 4(R0), #2336
```

```

: 0643
: 0707
:
```


51	01	00000424	8F	04	7F 13 00015	BEQL	7\$	:	0711
					A0 D1 00017	CMPL	4(R0), #1060	:	
					04 13 0001F	BEQL	1\$	:	
					FEFF 00021	BUGW		:	0713
					0000* 00023	.WORD	<BUG\$ UNXSIGNAL!4>	:	
		50		0000G	CF D0 00025	1\$:	IO PACKET, R0	:	0718
		51		20	A0 3C 0002A	MOVZWL	32(R0), FUNCTION	:	
		07			00 F0 0002E	INSV	#0, #7, #1, FUNCTION	:	
		00000072	8F		51 D1 00033	CMPL	FUNCTION, #114	:	0719
					09 13 0003A	BEQL	2\$	:	
		00000073	8F		51 D1 0003C	CMPL	FUNCTION, #115	:	0720
					0D 12 00043	BNEQ	3\$	:	
					50 DD 00045	2\$:	PUSHL R0	:	0722
					01 DD 00047	PUSHL	#1	:	
					5E DD 00049	PUSHL	SP	:	
				0000G	CF 9F 0004B	PUSHAB	ZERO CHANNEL	:	
		63			04 FB 0004F	CALLS	#4, SYSSCMKRNL	:	
		50		04	AC D0 00052	3\$:	MOVL SIGNAL, R0	:	0726
		52		08	A0 D0 00056	MOVL	8(R0), R2	:	
					05 13 0005A	BEQL	4\$	:	
		0000G	CF		52 B0 0005C	MOVW	R2, USER STATUS	:	0728
	0B	0B	AB		04 E0 00061	4\$:	BBS #4, 11(CURRENT_VCB), 5\$	:	0732
					7E D4 00066	CLRL	-(SP)	:	0734
					5E DD 00068	PUSHL	SP	:	
				0000V	CF 9F 0006A	PUSHAB	RETURN ALL ERR	:	
					03 FB 0006E	CALLS	#3, SYSSCMKRNL	:	
		00000224	8F		52 D1 00071	5\$:	CMPL R2, #548	:	0739
					0E 12 00078	BNEQ	6\$	:	
				38	AB DD 0007A	PUSHL	56(CURRENT_VCB)	:	0741
					01 DD 0007D	PUSHL	#1	:	
					5E DD 0007F	PUSHL	SP	:	
				0000G	CF 9F 00081	PUSHAB	TERMINATE VOL	:	
					04 FB 00085	CALLS	#4, SYSSCMKRNL	:	
					7E D4 00088	6\$:	CLRL -(SP)	:	0743
	7E	08	AC		08 C1 0008A	ADDL3	#8, MECHANISM, -(SP)	:	
		00000000G	9F		02 FB 0008F	CALLS	#2, @#SYSSUNWIND	:	
					04 00096	7\$:	RET	:	0746

; Routine Size: 151 bytes, Routine Base: \$CODE\$ + 00D6

; 367 0747 1

```
369 0748 1 GLOBAL ROUTINE RETURN_ALL_ERR : COMMON_CALL NOVALUE =
370 0749 1
371 0750 1 ++
372 0751 1
373 0752 1 FUNCTIONAL DESCRIPTION:
374 0753 1 This routine returns all blocked io with the error given in user_status
375 0754 1
376 0755 1 CALLING SEQUENCE:
377 0756 1 RETURN_ALL_ERROR, called in kernel mode
378 0757 1
379 0758 1 INPUT PARAMETERS:
380 0759 1 none
381 0760 1
382 0761 1 IMPLICIT INPUTS:
383 0762 1 CURRENT_VCB - address of current volume control block
384 0763 1
385 0764 1 OUTPUT PARAMETERS:
386 0765 1 none
387 0766 1
388 0767 1 IMPLICIT OUTPUTS:
389 0768 1 none
390 0769 1
391 0770 1 ROUTINE VALUE:
392 0771 1 none
393 0772 1
394 0773 1 SIDE EFFECTS:
395 0774 1 none
396 0775 1
397 0776 1 USER ERRORS:
398 0777 1 none
399 0778 1
400 0779 1 --
401 0780 1
402 0781 2 BEGIN
403 0782 2
404 0783 2 EXTERNAL REGISTER
405 0784 2 COMMON_REG;
406 0785 2
407 0786 2 EXTERNAL ROUTINE
408 0787 2 IO_DONE; ! complete io
409 0788 2
410 0789 2 EXTERNAL
411 0790 2 USER_STATUS; ! status to be returned to user
412 0791 2
413 0792 2 LOCAL
414 0793 2 STATUS, ! save status
415 0794 2 PACKET : REF BBLOCK;
416 0795 2
417 0796 2 STATUS = .USER_STATUS; ! save status
418 0797 2 USER_STATUS<16, 16> = 0; ! these io's have no transfer count
419 0798 2
420 0799 2 WHILE 1
421 0800 2 DO
422 0801 2 BEGIN
423 0802 2
424 0803 2 IF REMQUE(.CURRENT_VCB[VCB$L_BLOCKFL], PACKET)
425 0804 2 THEN
```


CNTRL
V04-000

K 2
16-Sep-1984 02:11:48
14-Sep-1984 12:46:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MTAACP.SRC]CNTRL.B32;1
Page 13
(5)

```
: 426      0805      3      EXITLOOP;
: 427      0806      3
: 428      0807      3      PACKET[IRP$V VIRTUAL] = 0;
: 429      0808      3      IO_DONE(.PACKET);
: 430      0809      3      END;
: 431      0810      3
: 432      0811      3      USER_STATUS = .STATUS;
: 433      0812      1      END;
                                ! make sure not seen again
                                ! restore status
                                ! end of routine
```

				000C 00000	.ENTRY	RETURN ALL ERR, Save R2,R3	: 0748
53	0000G	CF	D0	00002	MOVL	USER_STATUS, STATUS	: 0796
	0000G	CF	B4	00007	CLRW	USER_STATUS+2	: 0797
52	00	BB	0F	0000B	1\$: REMQUE	@0(CURRENT_VCB), PACKET	: 0803
		0D	1D	0000F	BVS	2\$	
2A	A2	10	8A	00011	BICB2	#16, 42(PACKET)	: 0807
		52	DD	00015	PUSHL	PACKET	: 0808
0000G	CF	01	FB	00017	CALLS	#1, IO_DONE	
		ED	11	0001C	BRB	1\$	: 0799
0000G	CF	53	D0	0001E	2\$: MOVL	STATUS, USER_STATUS	: 0811
			04	00023	RET		: 0812

; Routine Size: 36 bytes, Routine Base: \$CODE\$ + 016D

; 434 0813 1

```
: 436      0814 1 ROUTINE CANCEL : COMMON_CALL NOVALUE =
: 437      0815 1
: 438      0816 1 ++
: 439      0817 1
: 440      0818 1 FUNCTIONAL DESCRIPTION:
: 441      0819 1     This routine signals the cancel error. The call is necessary so
: 442      0820 1     that the unwind returns to the control module and not exit
: 443      0821 1
: 444      0822 1 CALLING SEQUENCE:
: 445      0823 1
: 446      0824 1 INPUT PARAMETERS:
: 447      0825 1     none
: 448      0826 1
: 449      0827 1 IMPLICIT INPUTS:
: 450      0828 1     none
: 451      0829 1
: 452      0830 1 OUTPUT PARAMETERS:
: 453      0831 1     none
: 454      0832 1
: 455      0833 1 IMPLICIT OUTPUTS:
: 456      0834 1     none
: 457      0835 1
: 458      0836 1 ROUTINE VALUE:
: 459      0837 1     none
: 460      0838 1
: 461      0839 1 SIDE EFFECTS:
: 462      0840 1     none
: 463      0841 1
: 464      0842 1 USER ERRORS:
: 465      0843 1     none
: 466      0844 1
: 467      0845 1 --
: 468      0846 1
: 469      0847 2 BEGIN
: 470      0848 2
: 471      0849 2 EXTERNAL REGISTER
: 472      0850 2     COMMON_REG;
: 473      0851 2
: 474      0852 2 ERR_EXIT(SS$_CANCEL);
: 475      0853 1 END;
```

```
0830 8F BF 00002 CANCEL: .WORD Save nothing
      04 00006          CHMU #2096
                        RET
```

```
: 0814
: 0852
: 0853
```

; Routine Size: 7 bytes, Routine Base: \$CODE\$ + 0191

```
: 476      0854 1 END
: 477      0855 1
: 478      0856 0 ELUDOM
```


PSECT SUMMARY									
Name	Bytes	Attributes							
\$CODE\$	408	NOVEC,NOWRT, RD	EXE,NOSHR,	LCL,	REL,	CON,NOPIC,ALIGN(2)			
\$LOCKEDD1\$	108	NOVEC, WRT, RD	NOEXE,NOSHR,	LCL,	REL,	CON,NOPIC,ALIGN(2)			
. ABS .	0	NOVEC,NOWRT,NORD	NOEXE,NOSHR,	LCL,	ABS,	CON,NOPIC,ALIGN(0)			

Library Statistics					
File	-----		Symbols	-----	
	Total	Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	31	0	1000	00:01.9

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:CNTRL/OBJ=OBJ\$:CNTRL MSRC\$:CNTRL/UPDATE=(ENH\$:CNTRL)

Size: 408 code + 108 data bytes

Run Time: 00:12.1

Elapsed Time: 00:46.1

Lines/CPU Min: 4248

Lexemes/CPU-Min: 18074

Memory Used: 121 pages

Compilation Complete

0254 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

