

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```
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
```

```
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)      ! end of routine startup
: 111     0493 1 END;

```

```

.TITLE CNTRL
.IDENT \V04-000\
.EXTRN SYSSCMEXEC
.PSECT $CODE$,NOWRT,2
.ENTRY STARTUP, Save nothing      : 0459
CLRL  -(SP)                        : 0492
PUSHL SP
PUSHAB MTA_CONTROL
CALLS #3, @#SYSSCMEXEC
RET
: 0493

```

```

0000 0000
7E D4 0002
5E DD 0004
0000000G 9F 0000V CF 9F 0006
03 FB 000A
04 0011

```

: 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
172 0553 2
173 0554 2
174 0555 2
175 0556 2
176 0557 2
177 0558 2
178 0559 2
179 0560 2
180 0561 2
181 0562 2
182 0563 2
183 0564 2
184 0565 2
185 0566 2
186 0567 2
187 0568 2
188 0569 2
189 0570 2
190 0571 2
191 0572 2
192 0573 2
193 0574 2
194 0575 2
195 0576 2
196 0577 2
197 0578 2
198 0579 2
199 0580 2
200 0581 2
201 0582 2
202 0583 2
203 0584 2
204 0585 2
205 0586 2
206 0587 2
207 0588 2
208 0589 2
209 0590 2
210 0591 2
211 0592 2
212 0593 2
213 0594 2
214 0595 2
215 0596 2
216 0597 2
217 0598 2
218 0599 2
219 0600 2
220 0601 2
221 0602 2
222 0603 2
223 0604 2
224 0605 2
225 0606 2
226 0607 2
227 0608 2

```

LOCAL FIB      : BBLOCK [FIB$K_MTALEN],      ! copy of fib
IMPURE_END    : VECTOR [0];

GLOBAL LITERAL
IMPURE_SIZE = IMPURE_END - USER_STATUS;

LOCAL
FUNCTION;      ! function code and qualifiers

OWN
RANGE      : VECTOR [2] INITIAL (0, %X'FFFFFFFF');

EXTERNAL ROUTINE
END_OF_VOL  : NOPRES,      ! end-of-volume processing
GET_REQ     : LSGET_REQ,   ! get next request
INIT_MTAACP,
IO_DONE,    ! acp initialization
MTA_ACCESS  : NOPRES,     ! complete i/o processing
MTA_ACPCNTRL : NOPRES,    ! access a file
MTA_CREATE  : NOPRES,     ! mtaacp control functions
MTA_DEACCESS : NOPRES,    ! create a file
MTA_MODIFY  : NOPRES,     ! deaccess a file
MTA_MOUNT   : NOPRES,     ! modify a file
SYSSPURGWS  : ADDRESSING_MODE (ABSOLUTE); ! mount a volume

! enable except_hndlr
!
BEGIN

BUILTIN
FP;

.FP = EXCEPT_HNDLR;
END;
KERNEL CALL(INIT_MTAACP);      ! initialize acp, one time only
SYSSPURGWS(RANGE);

WHILE 1
DO
BEGIN
! initialize impure area and set user status to success
!
CH$FILL(0, IMPURE_SIZE, USER_STATUS);
USER_STATUS[0] = T;
IO_PACKET = GET_REQ();      ! get a request
FUNCTION = .IO_PACKET[IRP$V_FCODE];

IF .CURRENT_VCB[VCBSV_CANCELIO] AND .FUNCTION NEQ IOS_DEACCESS
THEN
CANCEL()
ELSE
IF .FUNCTION EQL IOS_READPBLK OR .FUNCTION EQL IOS_WRITEPBLK
THEN
END_OF_VOL()      ! end of volume processing
ELSE

```

```

: 228 0609 4
: 229 0610 4
: 230 0611 4
: 231 0612 4
: 232 0613 4
: 233 0614 4
: 234 0615 4
: 235 0616 4
: 236 0617 4
: 237 0618 4
: 238 0619 4
: 239 0620 4
: 240 0621 4
: 241 0622 4
: 242 0623 4
: 243 0624 4
: 244 0625 4
: 245 0626 4
: 246 0627 4
: 247 0628 4
: 248 0629 4
: 249 0630 4
: 250 0631 4
: 251 0632 4
: 252 0633 4
: 253 0634 4
: 254 0635 4
: 255 0636 4
: 256 0637 4
: 257 0638 4
: 258 0639 4
: 259 0640 4
: 260 0641 1

```

```

BEGIN
CASE .FUNCTION FROM IOS_ACCESS TO IOS_MOUNT OF
SET
  [IOS_ACCESS] : MTA_ACCESS();
  [IOS_CREATE] : MTA_CREATE();
  [IOS_DEACCESS] : MTA_DEACCESS();
  [IOS_DELETE] : (ERROR(SS$_ILLIOFUNC);0);
  [IOS_MODIFY] : MTA_MODIFY();
  [IOS_ACPCONTROL] : MTA_ACPCTRL();
  [IOS_MOUNT] : MTA_MOUNT();
  [INRANGE] : (ERROR(SS$_ILLIOFUNC);0);
  [OUTRANGE] : (ERROR(SS$_ILLIOFUNC);0);
TES;
END;
IF .IO_PACKET NEQ 0
THEN
  KERNEL_CALL(IO_DONE, .IO_PACKET);
END;
END;
! end of while loop
! end of routine

```

```

.PSECT $LOCKEDD1$,NOEXE,2
0000 QUEUE_HEAD::
      .BLKB 4
0004 DISK_UCB::
      .BLKB 4
0008 IO_CHANNEL::
      .BLKB 4
000C MAIL_CHANNEL::
      .BLKB 4
0010 FREE_PAGE_HEAD::
      .BLKB 8
0018 LAST_PAGE::
      .BLKB 4
001C USER_STATUS::
      .BLKB 8
0024 IO_STATUS::
      .BLKB 8
002C IO_PACKET::
      .BLKB 4
0030 CURRENT_UCB::

```


						10\$-5\$,-		
						11\$-5\$		
						12\$		0630
0000G	CF	2A	11	00077		BRB		0614
		00	FB	00079	6\$:	CALLS	#0, MTA_ACCESS	
		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	12\$:	MOVZBW	#244, USER_STATUS	0628
	50	0000'	CF	D0	13\$:	MOVL	IO_PACKET, -R0	0635
			11	13		BEQL	14\$	
			50	DD		PUSHL	R0	0637
			01	DD		PUSHL	#1	
			5E	DD		PUSHL	SP	
00000000G	9F	0000G	CF	7F		PUSHAB	IO_DONE	
			04	FB		CALLS	#4, @#SYSS\$CMKRNL	
		FF60	31	000C1	14\$:	BRW	1\$	0589

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

: 261 0642 1

```

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 SSS_UNWIND which is

```


51	01	00000424	8F	04	7F 13 00015	BEQL	7\$		
					A0 D1 00017	CMPL	4(R0), #1060		0711
					04 13 0001F	BEQL	1\$		
					FEFF 00021	BUGW			0713
					0000* 00023	.WORD	<BUG\$ UNXSIGNAL!4>		
			50	0000G	CF D0 00025	1\$:	MOVL	ID 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	
			63		03 B 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	
			63		04 FB 00085		CALLS	#4, SYSSCMKRNL	
					7E D4 00088	6\$:	CLRL	-(SP)	0743
	7E	0B	AC		0B 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$BLOCKFL], PACKET)
425 0804 2 THEN

```

```

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

```

```

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

```

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

: 434 0813 1

```

: 436 0814 1 ROUTINE CANCEL · COMMON_CALL NGVALUE =
: 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,NOPI,ALIGN(2)
\$LOCKEDD1\$	108	NOVEC, WRT, RD, NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
. ABS .	0	NOVEC,NOWRT,NORD, NOEXE,NOSHR, LCL, ABS, CON,NOPI,ALIGN(0)

Library Statistics

file	----- Symbols -----		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$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

