


```

IIIIII  NN    NN  IIIIII  MM    MM  TTTTTTTTTT  AAAAAA
IIIIII  NN    NN  IIIIII  MM    MM  TTTTTTTTTT  AAAAAA
  II    NN    NN  II      MMMM  MMMM  TT          AA    AA
  II    NN    NN  II      MMMM  MMMM  TT          AA    AA
  II    NNNN  NN  II      MM   MM   TT          AA    AA
  II    NNNN  NN  II      MM   MM   TT          AA    AA
  II    NN  NN  NN  II      MM   MM   TT          AA    AA
  II    NN  NN  NN  II      MM   MM   TT          AA    AA
  II    NN    NNNN  II      MM   MM   TT          AA    AA
  II    NN    NNNN  II      MM   MM   TT          AA    AA
  II    NN    NN  II      MM   MM   TT          AA    AA
  II    NN    NN  II      MM   MM   TT          AA    AA
IIIIII  NN    NN  IIIIII  MM    MM  TT          AA    AA
IIIIII  NN    NN  IIIIII  MM    MM  TT          AA    AA

```

```

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
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```

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 INIMTA (LANGUAGE (BLISS32) ,
0003 0 IDENT = 'V04-000' ,
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0011 1 * ALL RIGHTS RESERVED. *
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0018 1 * TRANSFERRED. *
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0022 1 * CORPORATION. *
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1 ++
0031 1
0032 1 FACILITY: MTAACP
0033 1
0034 1 ABSTRACT:
0035 1 One time initialization for mtaACP
0036 1
0037 1 ENVIRONMENT:
0038 1
0039 1 Starlet operating system, including privileged system services
0040 1 and internal exec routines.
0041 1
0042 1 --
0043 1
0044 1
0045 1
0046 1 AUTHOR: D. H. GILLESPIE, CREATION DATE: 25-MAY-77 15:00
0047 1
0048 1 MODIFIED BY:
0049 1
0050 1 V02-005 DMW00076 David Michael Walp 8-Feb-1982
0051 1 Store away the ACP's username and account name
0052 1
0053 1 V02-004 DMW00013 David Michael Walp 14-Mar-1981
0054 1 Changed calculation of CCB address to GET_CCB
0055 1
0056 1 V02-003 KDM0037 Kathleen D. Morse 12-Feb-1981
0057 1 Change non-kernel mode references to SCH$GL_CURPCB to
```

```

58 0058 1 |         use CTL$GL_PCB instead.
59 J059 1 |
60 0060 1 |         V02-002 REFORMAT      Maria del C. Nasr      30-Jun-1980
61 0061 1 |
62 0062 1 |  !**
63 0063 1 |
64 0064 1 | LIBRARY 'SYS$LIBRARY:LIB.L32';
65 0065 1 |
66 0066 1 | REQUIRE 'SRC$:MTADEF.B32';
67 0450 1 |
68 0451 1 | ! Dummy vectors to bracket the locked down code and data psects.
69 0452 1 |
70 0453 1 |
71 0454 1 | PSECT
72 0455 1 |     OWN = $LOCKEDC$(NOWRITE, EXECUTE, ALIGN (9));
73 0456 1 |
74 0457 1 | OWN
75 0458 1 |     L_CODE_START           : VECTOR [0];
76 0459 1 |
77 0460 1 | PSECT
78 0461 1 |     OWN = $LOCKEDC9$(NOWRITE, EXECUTE, ALIGN (2));
79 0462 1 |
80 0463 1 | OWN
81 0464 1 |     L_CODE_END             : VECTOR [0];
82 0465 1 |
83 0466 1 | PSECT
84 0467 1 |     OWN = $LOCKEDDOS;
85 0468 1 |
86 0469 1 | OWN
87 0470 1 |     L_DATA_START          : VECTOR [0];
88 0471 1 |
89 0472 1 | PSECT
90 0473 1 |     OWN = $LOCKEDD9$;
91 0474 1 |
92 0475 1 | OWN
93 0476 1 |     L_DATA_END            : VECTOR [0];
94 0477 1 |
95 0478 1 | PSECT
96 0479 1 |     OWN = $LOCKEDD1$;
97 0480 1 |

```

IOD
Sym
ABD
ABD
ABD
ABD
ABD
ABD
AOB
CHE
FCB
FIB
IOS
IOS
IOC
IO
IPC
IRP
IRP
IRP
IRP
IRP
IRP
IRP
IRP
IRP
IRP
IRP
IRP
IRP
IRP
IRP
IRP
LOC
MVL
PAC
PR\$
RVT
UCB
UCB
USE
VCB
VCB
WCB
PSE

SAE
SCC

```

: 99      0481 1 GLOBAL ROUTINE INIT_MTAACP : NOVALUE =
100      0482 1
101      0483 1 |++
102      0484 1
103      0485 1 | FUNCTIONAL DESCRIPTION:
104      0486 1 |     One time initialization of mtaACP
105      0487 1
106      0488 1 | CALLING SEQUENCE:
107      0489 1 |     INIT_MTAACP(), in kernel mode
108      0490 1
109      0491 1 | INPUT PARAMETERS:
110      0492 1 |     none
111      0493 1
112      0494 1 | IMPLICIT INPUTS:
113      0495 1 |     IO data base
114      0496 1
115      0497 1 | OUTPUT PARAMETERS:
116      0498 1 |     none
117      0499 1
118      0500 1 | IMPLICIT OUTPUTS:
119      0501 1 |     IO_CHANNEL      : channel # for I/O
120      0502 1 |     MAIL_CHANNEL    : channel # for mailbox
121      0503 1 |     QUEUE_HEAD      : ACP queue list head
122      0504 1 |     FREE_PAGE_HEAD  : initialized to be empty
123      0505 1 |     LAST_PAGE       : 0 (not determined yet)
124      0506 1
125      0507 1 | ROUTINE VALUE:
126      0508 1 |     none
127      0509 1
128      0510 1 | SIDE EFFECTS:
129      0511 1 |     none
130      0512 1
131      0513 1 | --
132      0514 1
133      0515 2 BEGIN
134      0516 2
135      0517 2 EXTERNAL
136      0518 2     CTLST_USERNAME : ADDRESSING_MODE (ABSOLUTE),
137      0519 2     CTLST_ACCOUNT  : ADDRESSING_MODE (ABSOLUTE);
138      0520 2
139      0521 2 GLOBAL
140      0522 2 | store away the username and account name of the ACP
141      0523 2 | use the size of the Volume Virtual Page ( which has been checked
142      0524 2 | against the JIB definations
143      0525 2
144      0526 2     ACP_USERNAME : VECTOR [ VVPSS_USERNAME, BYTE ],
145      0527 2     ACP_ACCOUNT  : VECTOR [ VVPSS_ACCOUNT,  BYTE ];
146      0528 2
147      0529 2 LOCAL
148      0530 2     SYSEXV  : VECTOR [2],           ! sys*em exception handlers
149      0531 2     AQB      : REF BBLOCK,         ! pointer to scan AQB list
150      0532 2     CCB      : REF BBLOCK;         ! pointer to channel control block
151      0533 2
152      0534 2 LITERAL
153      0535 2     MSGSIZE = 128
154      0536 2     PROTECTION = %x'0000';
155      0537 2

```

```

156 0538 2
157 0539 2
158 0540 2
159 0541 2
160 0542 2
161 0543 2
162 0544 2
163 0545 2
164 0546 2
165 0547 2
166 0548 2
167 0549 2
168 0550 2
169 0551 2
170 0552 2
171 0553 2
172 0554 2
173 0555 2
174 0556 2
175 0557 2
176 0558 2
177 0559 2
178 0560 2
179 0561 2
180 0562 2
181 0563 2
182 0564 2
183 0565 2
184 0566 2
185 0567 2
186 0568 2
187 0569 2
188 0570 2
189 0571 2
190 0572 2
191 0573 2
192 0574 2
193 0575 2
194 0576 2
195 0577 2
196 0578 2
197 0579 2
198 0580 2
199 0581 2
200 0582 2
201 0583 2
202 0584 2
203 0585 2
204 0586 2
205 0587 2
206 0588 2
207 0589 2
208 0590 2
209 0591 2
210 0592 2
211 0593 2
212 0594 2

```

```

EXTERNAL
    IO_CHANNEL      : REF BBLOCK,      ! channel # for I/O
    DISK_UCB        : REF BBLOCK,      ! UCB address of 'sys$disk'
    IOCSGL_AQBLIST  : REF BBLOCK ADDRESSING_MODE (ABSOLUTE),
    FREE_PAGE_HEAD  : VECTOR [2],      ! address of two long words which are
                                         ! the free page list head
    LAST_PAGE       : REF BBLOCK,      ! last page in program region
    MAIL_CHANNEL    : REF BBLOCK,      ! channel # for mailbox
    QUEUE_HEAD      : REF BBLOCK,      ! address of queue head
    CTLSGL_PCB      : REF BBLOCK ADDRESSING_MODE (ABSOLUTE);
                                         ! address of current PCB

EXTERNAL ROUTINE
    SYSSASSIGN      : ADDRESSING_MODE (ABSOLUTE),      ! assign device
    SYSSCREMBX      : ADDRESSING_MODE (ABSOLUTE),      ! create a mailbox
    GET_CCB;        : ADDRESSING_MODE (ABSOLUTE),      ! get the address of the CCB

! disable the kernel exception vectors
$SETEXV(VECTOR = 0, ACMODE = EXEC_MODE, PRVHND = SYSEXV[0]);
$SETEXV(VECTOR = 1, ACMODE = EXEC_MODE, PRVHND = SYSEXV[1]);

! **** for debug: if the previous handler addresses are found to be in
! process space, put them back since they belong to the debugger

IF .SYSEXV[0] GTR 0
THEN
    $SETEXV(VECTOR = 0, ACMODE = EXEC_MODE, ADDRES = .SYSEXV[0]);

IF .SYSEXV[1] GTR 0
THEN
    $SETEXV(VECTOR = 1, ACMODE = EXEC_MODE, ADDRES = .SYSEXV[1]);

! Find the queue header for this ACP by searching the system AQB list
! for an AQB with a matching PID. Then assign a channel to 'sys$disk' (as
! good a device as any) and record the channel number. Said channel number
! is used for all I/O - it is assigned to the right device simply by
! stuffing the UCB pointer in the CCB. Also save the UCB address in the
! channel so that it can be restored to properly deassign the channel when
! we exit.
AQB = .IOCSGL_AQBLIST;
UNTIL .AQB EQL 0 OR .AQB[AQB$ACPPID] EQL .CTLSGL_PCB[PCB$PID]
DO
    AQB = .AQB[AQB$LINK];

IF .AQB NEQ 0
THEN
    QUEUE_HEAD = .AQB
ELSE
    $EXIT(CODE = SSS_NOAQB);

! Now lock some code and data into the working set. Some is necessary
! inorder to raise IPL.

```

```
213 0595 2 $ADJWSL(PAGCNT = -10000);           ! adjust working set to minimum
214 0596 2 $ADJWSL(PAGCNT = 64);             ! then add desired size
215 0597 2 (
216 0598 2
217 0599 2 LOCAL
218 0600 2     STATUS;
219 0601 2
220 0602 2 STATUS = $LKWSET(INADR = UPLIT (L_CODE_START, L_CODE_END));
221 0603 2
222 0604 2 IF NOT .STATUS
223 0605 2 THEN
224 0606 2     $EXIT(CODE = .STATUS));
225 0607 2
226 0608 2 BEGIN
227 0609 2
228 0610 2 LOCAL
229 0611 2     STATUS;
230 0612 2
231 0613 2 STATUS = $LKWSET(INADR = UPLIT (L_DATA_START, L_DATA_END));
232 0614 2
233 0615 2 IF NOT .STATUS
234 0616 2 THEN
235 0617 2     $EXIT(CODE = .STATUS);
236 0618 2
237 0619 2 END;
238 0620 2
239 0621 2 MAIL_CHANNEL = 0;
240 0622 2
241 0623 2 BEGIN
242 0624 2
243 0625 2 LOCAL
244 0626 2     STATUS;
245 0627 2
246 0628 2 STATUS = SYSS$CREMBX(0, MAIL_CHANNEL, MSGSIZE, 0, PROTECTION, EXEC_MODE, 0);
247 0629 2
248 0630 2 IF NOT .STATUS
249 0631 2 THEN
250 0632 2     $EXIT(CODE = .STATUS);
251 0633 2
252 0634 2 END;
253 0635 2
254 0636 2 ! setup free page header
255 0637 2 !
256 0638 2 FREE_PAGE_HEAD[0] = FREE_PAGE_HEAD;
257 0639 2 FREE_PAGE_HEAD[1] = FREE_PAGE_HEAD;
258 0640 2 LAST_PAGE = 0;
259 0641 2
260 0642 2 ! assign channel for I/O and create a mailbox to communicate with operator
261 0643 2 !
262 0644 2 IO_CHANNEL = 0;           ! clear upper word
263 0645 2
264 0646 2 IF NOT SYSS$ASSIGN(Descriptor('SYSS$SYSTEM'), IO_CHANNEL, EXEC_MODE, 0)
265 0647 2 THEN
266 0648 2     BUG_CHECK(NOACPCHAN);
267 0649 2
268 0650 2 CCB = GET_CCB ( .IO_CHANNEL );
269 0651 2 DISK_UCB = .CCB[CCB$L_UCB];
```

```

: 270 0652 2
: 271 0653 2
: 272 0654 2
: 273 0655 2
: 274 0656 2
: 275 0657 2
: 276 0658 2
: 277 0659 2
: 278 0660 2
: 279 0661 2
: 280 0662 2
: 281 0663 1

```

```

! Clear the transition bit in the AQB to tell MOUNT that we are alive and
! well.
QUEUE_HEAD[AQBSV_CREATING] = 0;
! stuff away the ACP's account and username ( used to restore from after a
! request is made to the operator )
CH$MOVE ( VVPSS_USERNAME, CTLST_USERNAME, ACP_USERNAME );
CH$MOVE ( VVPSS_ACCOUNT, CTLST_ACCOUNT, ACP_ACCOUNT );
END; ! end of routine

```

```

.TITLE INIMTA
.IDENT \V04-000\
.PSECT $CODE$,NOWRT,2
00000000' 00000000' 00000 P.AAA: .ADDRESS L_CODE_START, L_CODE_END
00000000' 00000000' 00008 P.AAB: .ADDRESS L_DATA_START, L_DATA_END
4D 45 54 53 59 53 24 53 59 53 00010 P.AAD: .ASCII \SYS$SYSTEM\
0001A .BLKB 2
0000000A 0001C P.AAC: .LONG 10
00000000' 00020 .ADDRESS P.AAD
.PSECT $LOCKEDD9$,NOEXE,2
00000 L_DATA_END:
.BLKB 0
.PSECT $LOCKEDD0$,NOEXE,2
00000 L_DATA_START:
.BLKB 0
.PSECT $LOCKEDC9$,NOWRT,2
00000 L_CODE_END:
.BLKB 0
.PSECT $LOCKEDC0$,NOWRT,9
00000 L_CODE_START:
.BLKB 0
.PSECT $LOCKEDD1$,NOEXE,2
00000 ACP_USERNAME::
.BLKB 12
0000C ACP_ACCOUNT::
.BLKB 8
.EXTRN CTLST_USERNAME, CTLST_ACCOUNT
.EXTRN IO_CHANNEL, DISK_UCB
.EXTRN IOCSGL_AQBLIST, FREE_PAGE_HEAD
.EXTRN LAST_PAGE, MAIL_CHANNEL

```



```

      .EXTRN  QUEUE HEAD, CTL$GL PCB
      .EXTRN  SYS$ASSIGN, SYS$CREMBX
      .EXTRN  GET CCB, SYS$SETEXV
      .EXTRN  SYS$EXIT, SYS$ADJWSL
      .EXTRN  SYS$LKWSÉT, BUG$_NOACPCHAN

      .PSECT  $CODES, NOWRT, 2

      OFFC 00000
      .ENTRY  INIT MTAACP, Save R2,R3,R4,R5,R6,R7,R8,R9,- R10,R11 ; 0481
5B      0000G  CF  9E 00002  MOVAB  IO CHANNEL, R11
5A      0000G  CF  9E 00007  MOVAB  FREE PAGE HEAD, R10
59      00000000G 00 9E 0000C  MOVAB  SYS$[KWSET, R9
58      00000000G 00 9E 00013  MOVAB  SYS$ADJWSL, R8
57      00000000G 00 9E 0001A  MOVAB  SYS$SETEXV, R7
56      00000000G 00 9E 00021  MOVAB  SYS$EXIT, R6
5E      08 C2 00028  SUBL2  #8, SP
      5E DD 0002B  PUSHL  SP
      01 DD 0002D  PUSHL  #1
      7E 7C 0002F  CLRQ   -(SP)
67      04 FB 00031  CALLS  #4, SYS$SETEXV
      04 AE 9F 00034  PUSHAB SY$EXV+4
      01 DD 00037  PUSHL  #1
7E      01 7D 00039  MOVQ   #1, -(SP)
67      04 FB 0003C  CALLS  #4, SYS$SETEXV
      6E D5 0003F  TSTL  SY$EXV
      0B 15 00041  BLEQ  1$
7E      01 7D 00043  MOVQ   #1, -(SP)
      08 AE DD 00046  PUSHL  SY$EXV
      7E D4 00049  CLRL  -(SP)
67      04 FB 0004B  CALLS  #4, SYS$SETEXV
      04 AE D5 0004E 1$: TSTL  SY$EXV+4
      0B 15 00051  BLEQ  2$
7E      01 7D 00053  MOVQ   #1, -(SP)
      0C AE DD 00056  PUSHL  SY$EXV+4
      01 DD 00059  PUSHL  #1
67      04 FB 0005B  CALLS  #4, SYS$SETEXV
51      00000000G 9F D0 0005E 2$: MOVL  @#IOCSGL_AQBLIST, AQB
      14 13 00065 3$: BEQL  4$
60      50 00000000G 9F D0 00C67  MOVL  @#CTL$GL_PCB, R0
      A0 0C A1 D1 0006E  CML  12(AQB), -96(R0)
      06 13 00073  BEQL  4$
51      10 A1 D0 00075  MOVL  16(AQB), AQB
      EA 11 00079  BRB   3$
      51 D5 0007B 4$: TSTL  AQB
      07 13 0007D  BEQL  5$
0000G  CF  51 D0 00C7F  MOVL  AQB, QUEUE_HEAD
      08 11 00084  BRB   6$
7E      0314 8F 3C 00086 5$: MOVZWL #788, -(SP)
66      01 FB 0008B  CALLS  #1, SYS$EXIT
      7E D4 0008E 6$: CLRL  -(SP)
7E      DBF0 8F 32 00090  CVTWL #-10000, -(SP)
68      02 FB 00095  CALLS  #2, SYS$ADJWSL
      7E D4 00098  CLRL  -(SP)
7E      40 8F 9A 0009A  MOVZBL #64, -(SP)
68      02 FB 0009E  CALLS  #2, SYS$ADJWSL
      7E 7C 000A1  CLRQ  -(SP)

```

0481
0557
0558
0564
0566
0568
0570
0580
0582
0584
0586
0588
0590
0595
0596
0602

69	FF35	CF	9F	000A3	PUSHAB	P,AAA		
05		03	FB	000A7	CALLS	#3, SYSS\$LKWSET		
		50	EB	000AA	BLBS	STATUS, 7\$	0604	
66		50	DD	000AD	PUSHL	STATUS	0606	
		01	FB	000AF	CALLS	#1, SYSEXIT		
		7E	7C	000B2	CLRQ	-(SP)	0613	
	FF2C	CF	9F	000B4	PUSHAB	P,AAB		
69		03	FB	000B8	CALLS	#3, SYSS\$LKWSET		
05		50	EB	000BB	BLBS	STATUS, 8\$	0615	
		50	DD	000BE	PUSHL	STATUS	0617	
66		01	FB	000C0	CALLS	#1, SYSEXIT		
	0000G	CF	D4	000C3	CLRL	MAIL_CHANNEL	0621	
7E		01	7D	000C7	MOVQ	#1, -(SP)	0628	
		7E	7C	000CA	CLRQ	-(SP)		
7E	80	8F	9A	000CC	MOVZBL	#128, -(SP)		
	0000G	CF	9F	000D0	PUSHAB	MAIL_CHANFL		
		7E	D4	000D4	CLRL	-(SP)		
00000000G	9F	07	FB	000D6	CALLS	#7, @#SYSS\$CREMBX		
	05	50	EB	000DD	BLBS	STATUS, 9\$	0630	
		50	DD	000E0	PUSHL	STATUS	0632	
66		01	FB	000E2	CALLS	#1, SYSEXIT		
6A		6A	9E	000E5	MOVAB	FREE_PAGE_HEAD, FREE_PAGE_HEAD	0638	
04	AA	6A	9E	000E8	MOVAB	FREE_PAGE_HEAD, FREE_PAGE_HEAD+4	0639	
	0000G	CF	D4	000EC	CLRL	LAST_PAGE	0640	
		6B	D4	000F0	CLRL	IO_CHANNEL	0644	
7E		01	7D	000F2	MOVQ	#1, -(SP)	0646	
		5B	DD	000F5	PUSHL	R11		
	00000000G	CF	9F	000F7	PUSHAB	P,AAC		
	9F	04	FB	000FB	CALLS	#4, @#SYSS\$ASSIGN		
	04	50	EB	00102	BLBS	RO, 10\$		
			FEFF	00105	BUGW		0648	
			0000*	00107	.WORD	<BUG\$ NOACPCHAN!4>		
	0000G	CF	6B	DD	PUSHL	IO_CHANNEL	0650	
	0000G	01	FB	00108	CALLS	#1, GET_CCB		
		60	D0	00110	MOVL	(CCB), DISK_UCB	0651	
		50	D0	00115	MOVL	QUEUE_HEAD, RO	0656	
	14	A0	08	8A	BICB2	#8, 20(RO)		
0000*	CF	00000000G	9F	0C	MOVQ3	#12, @#CTLST_USERNAME, ACP_USERNAME	0661	
0000*	CF	00000000G	9F	08	MOVQ3	#8, @#CTLST_ACCOUNT, ACP_ACCOUNT	0662	
			04	00132	RET		0663	

: Routine Size: 307 bytes, Routine Base: \$CODE\$ + 0024

```

: 282      0664  1
: 283      0665  1 END
: 284      0666  1
: 285      0667  0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
------	-------	------------

AGE
FCE
IOC
LOC
MVL
PR\$
RVT
SCH
SCH
SCH
UNL
VCE
WCE

PSE

\$AE
\$LC

Pha

Ini
Com
Pas
Sym
Pas
Sym
Pse
Crc
Ass

The
675
The
327
16

Mac

-S
-S
TO1

146

```

: $LOCKEDC0$      0 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(9)
: $LOCKEDC9$      0 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
: $LOCKEDD0$      0 NOVEC, WRT, RD , NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
: $LOCKEDD9$      0 NOVEC, WRT, RD , NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
: $LOCKEDD1$     20 NOVEC, WRT, RD , NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
: $CODE$         343 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

```

The
MAC

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	16 0	1000	00:01.9

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:INIMTA/OBJ=OBJ\$:INIMTA MSRCS:INIMTA/UPDATE=(ENHS:INIMTA)

```

: Size:          307 code + 56 data bytes
: Run Time:      00:10.3
: Elapsed Time: 00:23.1
: Lines/CPU Min: 3900
: Lexemes/CPU-Min: 23345
: Memory Used:  134 pages
: Compilation Complete

```

