M 5

```
MMM         MMM     AAAAAAAAA        CCCCCCCCCCCC  RRRRRRRRRRRR        OOOOOOOOO
MMM         MMM     AAAAAAAAA        CCCCCCCCCCCC  RRRRRRRRRRRR        OOOOOOOOO
MMM         MMM     AAAAAAAAA        CCCCCCCCCCCC  RRRRRRRRRRRR        OOOOOOOOO
MMMMMM   MMMMMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMMMMM   MMMMMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMMMMM   MMMMMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMM   MMM   MMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMM   MMM   MMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMM   MMM   MMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRRRRRRRRRRR     OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRRRRRRRRRRR     OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRRRRRRRRRRR     OOO         OOO
MMM         MMM  AAAAAAAAAAAAAAA  CCC              RRR   RRR        OOO         OOO
MMM         MMM  AAAAAAAAAAAAAAA  CCC              RRR   RRR        OOO         OOO
MMM         MMM  AAAAAAAAAAAAAAA  CCC              RRR   RRR        OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRR      RRR     OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRR      RRR     OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRR      RRR     OOO         OOO
MMM         MMM  AAA         AAA     CCCCCCCCCCCC  RRR         RRR     OOOOOOOOO
MMM         MMM  AAA         AAA     CCCCCCCCCCCC  RRR         RRR     OOOOOOOOO
MMM         MMM  AAA         AAA     CCCCCCCCCCCC  RRR         RRR     OOOOOOOOO
```

```
_$
Sy
---
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(

MA(
MA(
MA(
MA(

MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
MA(
**FILE**ID**APSECT                                         K 9
```

```
  AAAAAA    PPPPPPPP      SSSSSSSS  EEEEEEEEEE    CCCCCCCC  TTTTTTTTTT
  AAAAAA    PPPPPPPP      SSSSSSSS  EEEEEEEEEE    CCCCCCCC  TTTTTTTTTT
AA      AA  PP      PP  SS          EE          CC              TT
AA      AA  PP      PP  SS          EE          CC              TT
AA      AA  PP      PP  SS          EE          CC              TT
AA      AA  PP      PP  SS          EE          CC              TT
AA      AA  PPPPPPPP      SSSSSS    EEEEEEEE    CC              TT
AA      AA  PPPPPPPP      SSSSSS    EEEEEEEE    CC              TT
AAAAAAAAAA  PP                  SS  EE          CC              TT
AAAAAAAAAA  PP                  SS  EE          CC              TT
AA      AA  PP                  SS  EE          CC              TT          ....
AA      AA  PP                  SS  EE          CC              TT          ....
AA      AA  PP          SSSSSSSS    EEEEEEEEEE    CCCCCCCC      TT          ....
AA      AA  PP          SSSSSSSS    EEEEEEEEEE    CCCCCCCC      TT          ....

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
```

```
MA
Sy

$C
AL
AL
AL
AL
AR
AU
BL
CH
CH
CH
CH
CH
CH
CH
CH
CH
CH
CH
CH
CH
CH
CH
CH
CN
CR
CR
CS
EN
ER
FF
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
FL
```

Table of contents

```
0000          .TITLE  MAC$APSECT      PROCESS PSECT RELATED DIRECTIVES
0000          .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY:     VAX MACRO ASSEMBLER OBJECT LIBRARY
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ; The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000    ; modules for input to the VAX-11 LINKER.
0000    ;
0000    ; ENVIRONMENT:  USER MODE
0000    ;
0000    ; AUTHOR: Benn Schreiber, CREATION DATE: 20-AUG-78
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03.01  RRB030  Rowland R. Bradley      06-Jul-1984
0000    ;               Fix broken branch.
0000    ;
0000    ;       V01.10  RN0023          R. Newland      3-Nov-1979
0000    ;               New message codes to get error messages from system
0000    ;               message file.
0000    ;
0000    ;       V01.09  RN0013          R. Newland      27-Sep-1979
0000    ;               Use new symbols for PSECT option processing
0000    ;
0000    ;       V01.08  RN0005          R. Newland      12-Aug-1979
0000    ;               Variable symbol names and remove .ALIGN LONG statements
0000    ;
0000    ;       V01.07  008             B. Schreiber            23-JAN-1979
0000    ;               Clear PSECT Block before using it.
```

```
                                       0000    58 ;--
```

```
    0000               .SBTTL  DECLARATIONS
    0000       ;
    0000       ; INCLUDE FILES:
    0000       ;
    0000
    0000       ;
    0000       ; MACROS:
    0000       ;
    0000
    0000               $MAC_CTLFLGDEF                  ;DEFINE CONTROL FLAGS
    0000               $MAC_GENVALDEF                  ;DEFINE GENERAL VALUES
    0000               $MAC_INTCODDEF                  ;DEFINE INT. BUFFER CODES
    0000               $MAC_SYMBLKDEF                  ;DEFINE SYMBOL/PSECT BLOCK OFFSETS
    0000               $MACMSGDEF                      ; Define message codes
    0000
    0000       ;
    0000       ; EQUATED SYMBOLS:
    0000       ;
    0000
    0000       ;
    0000       ; OWN STORAGE:
    0000       ;
    0000
00000000               .PSECT  MAC$RO_CODE_P1,NOWRT,GBL,LONG
```

```
                          0000          .SBTTL  PSECT   PROCESS .PSECT STATEMENT
                          0000
                          0000  ;++
                          0000  ; FUNCTIONAL DESCRIPTION:
                          0000  ;
                          0000  ;       THIS ROUTINE FULLY PROCESSES THE .PSECT STATEMENT.  THE
                          0000  ;       OPTIONS ARE PARSED AND A PSECT BLOCK IS ALLOCATED AND
                          0000  ;       FILLED IN.
                          0000  ;
                          0000  ;--
                          0000
                          0000  PSECT::                                         ;DIRECTIVE = KPSECT
      03 0005'CF   E8     0000          BLBS    W^ENB$G_LOCALSYMB+SYM$L_VAL,10$ ;BRANCH IF ENABLE LSB
             FFF8'   30   0005          BSBW    MAC$SET_NEW_LSB                 ;NO--MAKE A NEW LSB NOW
             FFF5'   30   0008  10$:    BSBW    MAC$SET_PC                      ;RECORD HIGH PC
             FFF2'   30   000B          BSBW    MAC$SYMSCNUP                    ;GET THE PSECT NAME
           6A 50     E9   000E          BLBC    R0,50$                          ;BRANCH IF NO PSECT NAME
     56   0000'CF    9E   0011          MOVAB   W^MAC$GL_PSC_LIST,R6            ;YES--POINT TO PSECT LIST
          52   56    D0   0016          MOVL    R6,R2                           ;COPY IN CASE NOTHING IN LIST
          55   66    D0   0019          MOVL    SYM$L_LINK(R6),R5               ; Get next PSECT entry
               09    13   001C          BEQL    20$                             ;BRANCH IF NO PSECTS YET
             FFDF'   30   001E          BSBW    MAC$SRC_LIST                    ;SEE IF PSECT ALREADY DECLARED
          56   51    D0   0021          MOVL    R1,R6                           ;COPY RESULT IF FOUND
             59 50   E8   0024          BLBS    R0,60$                          ;BRANCH IF FOUND
               52    DD   0027  20$:    PUSHL   R2                              ;STACK ADDRESS OF PREVIOUS
     50   0000'CF    D0   0029          MOVL    W^MAC$GL_PSC_BLKP,R0            ;GET POINTER TO PSECT 2K BLOCK
               12    12   002E          BNEQ    40$                             ;IF NEQ GO USE IT
             FFCD'   30   0030  30$:    BSBW    MAC$ALL_2_PAGES                 ;ALLOCATE TWO PAGES
     0000'CF   50    D0   0033          MOVL    R0,W^MAC$GL_PSC_BLKP            ;SAVE ADDRESS FOR LATER
     60    03F8 8F   3C   0038          MOVZWL  #<1024-8>,(R0)                  ; Set # bytes in first longword
    04 A0    08 A0   9E   003D          MOVAB   8(R0),4(R0)                     ;SET 2ND LONGWORD AS POINTER TO FREE
     51   0000'CF    9A   0042  40$:    MOVZBL  W^MAC$AB_TMPSYM,R1              ; Get size of name
               51    D6   0047          INCL    R1                              ; Include count byte
          60   51    C2   0049          SUBL2   R1,(R0)                         ; Subtract from bytes available
          80   13    C2   004C          SUBL2   #PSC$K_BLKSIZ,(R0)+             ; and subtract fixed part size
               DF    19   004F          BLSS    30$                             ;IF LSS NO--GO ALLOCATE ANOTHER
          56   60    D0   0051          MOVL    (R0),R6                         ;GET POINTER TO PSECT BLOCK
          60   51    C0   0054          ADDL2   R1,(R0)                         ; Set pointer to end of allocated
          60   13    C0   0057          ADDL2   #PSC$K_BLKSIZ,(R0)              ; block
          7E   51    90   005A          MOVB    R1,-(SP)                        ; Save total length
 66   0000'CF  51    28   005D          MOVC3   R1,W^MAC$AB_TMPSYM,(R6)         ; Copy symbol count/name
          56   53    D0   0063          MOVL    R3,R6                           ; Save pointer to block
63  13  00  6E   00  2C   0066          MOVC5   #0,(SP),#0,#PSC$K_BLKSIZ,(R3) ; Clear rest of block
       04 A6   8E    90   006C          MOVB    (SP)+,SYM$B_NAME(R6)            ; Store offset to name
            52 8ED0       0070          POPL    R2                              ;GET POINTER TO PREVIOUS PSECT BLOCK
          66   62    D0   0073          MOVL    SYM$L_LINK(R2), -               ; Link in new PSECT block
                          0076                  SYM$L_LINK(R6)
          62   56    D0   0076          MOVL    R6,SYM$L_LINK(R2)               ; ...
               05    11   0079          BRB     60$
     56   0000'CF    9E   007B  50$:    MOVAB   W^PSECT$BLANK,R6                ;USE THE BLANK PSECT
  09 A6   0080 8F    A8   0080  60$:    BISW2   #SYM$M_REF,PSC$W_FLAG(R6) ;FLAG PSECT AS REFERENCED
              001A   30   0086          BSBW    MAC$PSC_OPT_SCN                 ;SCAN PSECT OPTIONS
                          0089          $INTOUT_LW INT$_PSECT,R6                ;SWITCH TO NEW PSECT IN PASS 2
  0000'CF    05 A6   D0   0091          MOVL    PSC$L_MAXLGTH(R6),W^MAC$GL_PC ;SET NEW PC
  0000'CF    0C A6   9A   0097          MOVZBL  PSC$B_SEG(R6),W^MAC$GL_PSECT ;AND NEW PSECT
     0000'CF   56    D0   009D          MOVL    R6,W^MAC$GL_PSECTPTR
                     05   00A2          RSB
```

```
                              00A3          .SBTTL  SCAN PSECT OPTIONS
                              00A3
                              00A3  MAC$PSC_OPT_SCN:
                    57    DD  00A3          PUSHL   R7                      ;SAVE R7
         0C 09 A6   00    E1  00A5          BBC     #SYM$V_DEF,SYM$W_FLAG(R6),10$ ;BRANCH IF PSECT NOT DEFINED
       57    0D A6  3C        00AA          MOVZWL  PSC$W_OPTIONS(R6),R7    ;DEFINED---GET DEFINED OPTIONS
            7E  57  D0        00AE          MOVL    R7,-(SP)                ;COPY TO -(SP) FOR POSITIVE OPTIONS
            7E  57  D2        00B1          MCOML   R7,-(SP)                ;COPY COMPLEMENTED OPTIONS TO -(SP)
                09  11        00B4          BRB     20$
   57    01C8 8F    3C        00B6  10$:    MOVZWL  #PSC$M_DEFAULT,R7       ;NEW PSECT--SET DEFAULT OPTIONS
                7E  D4        00BB          CLRL    -(SP)                   ;CLEAR POS. AND NEG. OPTIONS
                7E  D4        00BD          CLRL    -(SP)                   ;...
                              00BF  20$:
                              00BF
                              00BF  OPTION_SCAN:
        0D      5A  91        00BF          CMPB    R10,#CR                 ;END OF LINE?
                03  12        00C2          BNEQ    10$                     ;IF NEQ NO
              00AF  31        00C4  5$:     BRW     130$                    ;YES--EXIT
             FF36'  30        00C7  10$:    BSBW    MAC$SKIPSP              ;SKIP SPACES
        2C      5A  91        00CA          CMPB    R10,#^A/,/              ;SCAN TO A COMMA?
                08  12        00CD          BNEQ    20$                     ;IF NEQ NO
             FF2E'  30        00CF          BSBW    MAC$GETCHR              ;YES--SKIP IT
             FF2B'  30        00D2          BSBW    MAC$SKIPSP              ;SKIP SPACES AGAIN
                E8  11        00D5          BRB     OPTION_SCAN             ;CK FOR EOL AGAIN
        0D      5A  91        00D7  20$:    CMPB    R10,#CR                 ;GET TO EOL?
                E8  13        00DA          BEQL    5$                      ;IF EQL YES--ALL DONE
                04  E1        00DC  25$:    BBC     #CHR$V_NUM_BER,-        ;BRANCH IF NOT NUMERIC
   34 0000'CA                 00DE                  W^MAC$AB_CMSK_TAB(R10),60$
   08 6E        0E  E1        00E2          BBC     #PSC$V_ALIGNFLG,(SP),30$ ;YES--WAS ALIGNMENT SEEN?
              00CC  30        00E6          BSBW    OPTION_CONFLICT         ;YES--REPORT CONFLICT
             FF14'  30        00E9          BSBW    MAC$SKP_OPR             ;SKIP TO COMMA OR EOL
                D1  11        00EC          BRB     OPTION_SCAN             ;AND SCAN NEXT OPTION
             FF0F'  30        00EE  30$:    BSBW    MAC$DNUMBER             ;ACCUMULATE NUMERIC ALIGNMENT
   55     0000'CF   D0        00F1          MOVL    W^MAC$GL_VALUE,R5       ;GET ALIGNMENT SCANNED
        09      55  D1        00F6          CMPL    R5,#9                   ;LEGAL ALIGNMENT?
                0B  1B        00F9          BLEQU   40$                     ;IF LEQU YES
        55      09  9A        00FB          MOVZBL  #9,R5                   ;NO--USE PAGE ALIGNMENT
                              00FE          $MAC_ERR INVALIGN               ; Get error message code
             FEFA'  30        0103          BSBW    MAC$ERRORLN             ;REPORT ERROR TO PASS 2
   55   55      0A  78        0106  40$:    ASHL    #PSC$V_ALIGNMENT,R5,R5  ; Position alignment
    00  55      0E  E3        010A          BBCS    #PSC$V_ALIGNFLG,R5,50$  ;SET ALIGNMENT FLAG
        57      55  C8        010E  50$:    BISL2   R5,R7                   ;SET IN OPTIONS
        6E      55  C8        0111          BISL2   R5,(SP)                 ;SET IN COMPL. OPTIONS FOR CHECK
                A9  11        0114          BRB     OPTION_SCAN             ;CONTINUE SCANNING
                              0116  ;
                              0116  ; NOT A NUMBER
                              0116  ;
             FEE7'  30        0116  60$:    BSBW    MAC$SYMSCNUP            ;SCAN THE OPTION NAME
             0D 50  E8        0119          BLBS    R0,70$                  ;BRANCH IF OPTION SCANNED
                              011C          $MAC_ERR DIRSYNX                ; No--get message code
             FEDC'  30        0121          BSBW    MAC$ERRORLN             ;REPORT ERROR TO PASS 2
             FED9'  30        0124          BSBW    MAC$SKP_OPR             ;SKIP TO COMMA OR EOL
                96  11        0127          BRB     OPTION_SCAN             ;SCAN NEXT OPTION
                              0129  ;
                              0129  ; LOOK UP THE OPTION
                              0129  ;
   55     0000'CF   9E        0129  70$:    MOVAB   W^PSC$G_OPTIONS,R5      ;POINT TO THE OPTIONS
```

```
            FECF'   30  012E            BSBW    MAC$SRC_LIST            ;LOOK IT UP
          0B 50     E8  0131            BLBS    R0,80$                  ;BRANCH IF FOUND
                        0134            $MAC_ERR NOTPSECOPT             ; No--get error code
            FEC4'   30  0139            BSBW    MAC$ERRORLN             ;ISSUE ERROR TO PASS 2
             FF80   31  013C            BRW     OPTION_SCAN             ;SCAN NEXT OPTION
     55   05 A1     32  013F    80$:    CVTWL   SYM$L_VAL(R1),R5        ;GET BITS FOR OPTION
               16   14  0143            BGTR    100$                    ;IF GTR NOT COMPLEMENT OF OPTION
        55     55   D2  0145            MCOML   R5,R5                   ;COMPLEMENTED--GET UNCOMPLEMENTED
     04 AE     55   D3  0148            BITL    R5,4(SP)                ;POSITIVE SET?
               05   13  014C            BEQL    90$                     ;IF EQL NO
             0064   30  014E            BSBW    OPTION_CONFLICT         ;YES--ISSUE MESSAGE
               20   11  0151            BRB     120$                    ;CONTINUE SCAN
        57     55   CA  0153    90$:    BICL2   R5,R7                   ;CLEAR OPTION
        6E     55   C8  0156            BISL2   R5,(SP)                 ;SET IN NEGATIVE MASK
               18   11  0159            BRB     120$                    ;CONTINUE SCANNING
                        015B    ;
                        015B    ; POSITIVE SENSE OPTION
                        015B    ;
        6E     55   D3  015B    100$:   BITL    R5,(SP)                 ;WAS COMPLEMENT SET?
               05   13  015E            BEQL    110$                    ;IF EQL NO
             0052   30  0160            BSBW    OPTION_CONFLICT         ;YES--REPORT ERROR
               0E   11  0163            BRB     120$                    ;CONTINUE SCANNING
        57     55   C8  0165    110$:   BISL2   R5,R7                   ;SET IN OPTIONS WORD
     04 AE     55   C8  0168            BISL2   R5,4(SP)                ;AND IN POSITIVE MASK
     03 55     0E   E1  016C            BBC     #PSC$V_ALIGNFLG,R5,120$ ;IS THIS AN ALIGNMENT OPTION?
        6E     55   C8  0170            BISL2   R5,(SP)                 ;YES--SET IN NEG. MASK
             FF49   31  0173    120$:   BRW     OPTION_SCAN             ;CONTINUE SCANNING OPTIONS
                        0176    ;
                        0176    ; DONE SCANNING OPTIONS
                        0176    ;
  33 09 A6     00   E0  0176    130$:   BBS     #SYM$V_DEF,PSC$W_FLAG(R6),150$ ;BRANCH IF NOT NEW PSECT
 09 A6    0081 8F   A8  017B            BISW2   #SYM$M_DEF!SYM$M_REF,PSC$W_FLAG(R6) ;NEW--MARK DEFINED
                        0181                                            ;AND REFERENCED
     0D A6     57   B0  0181            MOVW    R7,PSC$W_OPTIONS(R6)    ;SET THE OPTIONS
                        0185            $INTOUT_LW INT$_NEWP,R6         ;DEFINE NEW PSECT IN PASS 2
          0000'CF   D6  018D            INCL    W^MAC$GL_PSC_MAX        ;COUNT ANOTHER PSECT
0100 8F   0000'CF   B1  0191            CMPW    W^MAC$GL_PSC_MAX,#256   ;DO WE HAVE TOO MANY PSECTS?
               08   1B  0198            BLEQU   140$                    ;IF LEQU NO
                        019A            $MAC_ERR TOOMNYPSEC             ; Yes--get code
            FE5E'   30  019F            BSBW    MAC$ERRORLN             ;SEND TO PASS 2
 0C A6    0000'CF   90  01A2    140$:   MOVB    W^MAC$GL_PSC_MAX,PSC$B_SEG(R6) ;SET THE PSECT NUMBER
            05 A6   D4  01A8            CLRL    PSC$L_MAXLGTH(R6)       ;CLEAR MAX LENGTH
            0F A6   D4  01AB            CLRL    PSC$L_CURLOC(R6)        ;START AT 0
        5E     08   C0  01AE    150$:   ADDL2   #2*4,SP                 ;CLEAR TWO WORDS FROM STACK
             57 8ED0    01B1            POPL    R7                      ;RESTORE R7
                   05   01B4            RSB
                        01B5
                        01B5    OPTION_CONFLICT:
                        01B5            $MAC_ERR PSECOPCNFL             ; Get message code
            FE43'   31  01BA            BRW     W^MAC$ERRORLN           ;ISSUE MESSAGE AND RETURN
```

```
                                01BD                  .SBTTL  PROCESS .SAVE DIRECTIVE
                                01BD
                                01BD      ;++
                                01BD      ; FUNCTIONAL DESCRIPTION:
                                01BD      ;
                                01BD      ;           CSAVE IS CALLED TO PROCESS THE .SAVE DIRECTIVE.  THE
                                01BD      ;           NUMBER OF THE CURRENT PSECT IS SAVED IN THE PSECT
                                01BD      ;           SAVE BUFFER.  THE PC FOR THE CURRENT PSECT IS SAVED
                                01BD      ;           IN PSC$L_CURLOC OF THE CURRENT PSECT'S PSECT SYMBOL BLOCK.
                                01BD      ;
                                01BD      ;--
                                01BD
                                01BD      CSAVE::                                               ;DIRECTIVE = KSAVE
          56     0000'CF   9A   01BD                  MOVZBL  W^MAC$GL_PSC_SBP,R6       ;GET BUFFER INDEX
               1F     56   91   01C2                  CMPB    R6,#31                    ;OVERFLOW?
                      0B   1B   01C5                  BLEQU   10$                       ;IF LEQU NO
                                01C7                  $MAC_ERR PSECBUFOVF               ; Yes--get message cod.
                 5A   0D   D0   01CC                  MOVL    #CR,R10                   ;FORCE END OF LINE
                   FE2E'   31   01CF                  BRW     MAC$ERRORLN               ;ISSUE AND RETURN
 00000000'E6     0000'CF   90   01D2      10$:        MOVB    W^MAC$GL_PSECT,L^MAC$AB_PSC_SBF(R6) ;SAVE PSECT NUMBER
                 0000'CF   D6   01DB                  INCL    W^MAC$GL_PSC_SBP          ;BUMP THE INDEX
          52     0000'CF   D0   01DF                  MOVL    W^MAC$GL_PSECTPTR,R2      ;GET POINTER TO PSECT BLOCK
       0F A2     0000'CF   D0   01E4                  MOVL    W^MAC$GL_PC,PSC$L_CURLOC(R2) ;SAVE CURRENT PC
                                01EA                  $INTOUT_LW INT$_SAVE,R2           ;ISSUE SAVE TO PASS 2
                   FE0B'   30   01F2                  BSBW    MAC$SYMSCNUP              ;SEE IF THERE IS AN ARGUMENT
                  0D 50    E8   01F5                  BLBS    R0,30$                    ;BRANCH IF THERE WAS
                 0D   5A   91   01F8                  CMPB    R10,#CR                   ;NO--BUT DID WE GET TO EOL?
                      1F   13   01FB                  BEQL    40$                       ;IF EQL YES
                                01FD      20$:        $MAC_ERR DIRSYNX                  ; No--call it directive syntax
                   FDFB'   31   0202                  BRW     MAC$ERRORLN               ;REPORT THE ERROR
          55     0000'CF   9E   0205      30$:        MOVAB   W^MAC$G_LSBNAM,R5         ;POINT TO LSB NAMES
                   FDF3'   30   020A                  BSBW    MAC$SRC_LIST              ;SEE IF IT IS LEGAL FOR LSB
                  ED 50    E9   020D                  BLBC    R0,20$                    ;BRANCH IF ILL ARG
00000000'EF46    0000'CF   D0   0210                  MOVL    W^MAC$GL_LSB,L^MAC$AL_PSC_SLB[R6] ;YES--SAVE LSB
                      07   11   021A                  BRB     50$                       ;AND EXIT
        00000000'EF46      D4   021C      40$:        CLRL    L^MAC$AL_PSC_SLB[R6]      ;ENSURE NO LSB SAVED
                           05   0223      50$:        RSB
```

```
                              0224     289             .SBTTL  PROCESS .RESTORE DIRECTIVE
                              0224     290
                              0224     291 ;++
                              0224     292 ; FUNCTIONAL DESCRIPTION:
                              0224     293 ;
                              0224     294 ;       CRESTO IS CALLED TO PROCESS THE .RESTORE PSECT DIRECTIVE.
                              0224     295 ;       CODE IS EMITTED TO PASS 2 TO SWITCH PSECTS.
                              0224     296 ;
                              0224     297 ;--
                              0224     298
                              0224     299 CRESTO::                                        ;DIRECTIVE = KRESTORE
              FDD9'     30    0224     300             BSBW    MAC$SET_PC                  ;RECORD HIGH PC
     56   0000'CF       D0    0227     301             MOVL    W^MAC$GL_PSC_SBP,R6         ;GET SAVE BUFFER INDEX
                 09     14    022C     302             BGTR    10$                         ;IF GTR OK
                              022E     303             $MAC_ERR PSECBUFUND                 ; No--get error code
              FDCA'     30    0233     304             BSBW    MAC$ERRORPT                 ;REPORT TO PASS 2
                        05    0236     305             RSB                                 ;RETURN
                 56     D7    0237     306 10$:        DECL    R6                          ;BACK UP THE INDEX
        0000'CF  56     D0    0239     307             MOVL    R6,W^MAC$GL_PSC_SBP         ;SAVE INDEX
   50   00000000'E6     9A    023E     308             MOVZBL  L^MAC$AB_PSC_SBF(R6),R0     ;GET THE PSECT INDEX
           01    50     91    0245     309             CMPB    R0,#1                       ;BLANK PSECT?
                 07     12    0248     310             BNEQ    20$                         ;IF NEQ NO
     55   0000'CF       9E    024A     311             MOVAB   W^PSECT$BLANK,R5            ;YES--SET FOR IT
                 14     11    024F     312             BRB     50$                         ;
     55   0000'CF       D0    0251     313 20$:        MOVL    W^MAC$GL_PSC_LIST,R5        ;POINT TO PSECT LIST
                 36     13    0256     314             BEQL    MAC$PSC_RES_ERR             ;IF EQL NOTHING THERE
       0C A5     50     91    0258     315 30$:        CMPB    R0,PSC$B_SEG(R5)            ;IS THIS THE RIGHT PSECT?
                 07     13    025C     316             BEQL    50$                         ;IF EQL YES
           55    65     D0    025E     317             MOVL    (R5),R5                     ;NO--LINK TO NEXT
                 F5     12    0261     318             BNEQ    30$                         ;IF NEQ KEEP LOOKING
                 29     11    0263     319             BRB     MAC$PSC_RES_ERR             ;PSECT RESTORE ERROR
                              0265     320 50$:        $INTOUT_LW INT$_REST,R5             ;ISSUE RESTORE TO PASS 2
        0000'CF  55     D0    026D     321 60$:        MOVL    R5,W^MAC$GL_PSECTPTR        ;SET NEW PSECT POINTER
     0000'CF   0C A5    9A    0272     322             MOVZBL  PSC$B_SEG(R5),W^MAC$GL_PSECT ;AND NEW PSECT NUMBER
     0000'CF   0F A5    D0    0278     323             MOVL    PSC$L_CURLOC(R5),W^MAC$GL_PC ;SET NEW PC
 50   00000000'EF46     D0    027E     324             MOVL    L^MAC$AL_PSC_SLB[R6],R0     ;GET THE SAVED LSB NUMBER
                 05     13    0286     325             BEQL    70$                         ;IF EQL NO SAVED PSECT NUMBER
        0000'CF  50     D0    0288     326             MOVL    R0,W^MAC$GL_LSB             ;SET NEW LSB
                        05    028D     327 70$:        RSB
                              028E     328
                              028E     329 MAC$PSC_RES_ERR::
        0000'CF  00     FB    028E     330             CALLS   #0,W^MAC$ERR_INTERN         ;REPORT INTERNAL DIFFICULTIES
              FD6A'     31    0293     331             BRW     MAC$LAST_CHANCE             ;STAY TUNED FOR QUICK EXIT
```

```
                            0296   333          .SBTTL  ALIGNMENT DIRECTIVE
                            0296   334
                            0296   335 ;++
                            0296   336 ; FUNCTIONAL DESCRIPTION:
                            0296   337 ;
                            0296   338 ;        THIS DIRECTIVE CAUSES THE PC TO BE ALIGNED TO THE SPECIFIED
                            0296   339 ;        BOUNDARY, WITH THE OPTION OF THE SKIPPED BYTES BEING FILLED
                            0296   340 ;        WITH A FILL EXPRESSION.
                            0296   341 ;
                            0296   342 ;--
                            0296   343
                            0296   344 ALIGN::                                           ;ALSIGN_HEAD = KALIGN
       00 6B    06     E5   0296   345          BBCC    #FLG$V_EVALEXPR,(R11),.+1  ;DON'T OUTPUT EXPRESSION
       00 6B    07     E3   029A   346          BBCS    #FLG$V_EXPOPT,(R11),.+1 ;ASSUME NICE EXPRESSION
          0000'CF      D4   029E   347          CLRL    W^MAC$GL_ABSFLAG           ;ASSUME ABSOLUTE
             FD5B'     30   02A2   348          BSBW    MAC$SKIPSP                 ;SKIP SPACES
                04     E1   02A5   349          BBC     #CHR$V_NUM_BER,-           ;BRANCH IF NOT NUMERIC
       0E 0000'CA           02A7   350                  W^MAC$AB_CMSK_TAB(R10),30$
             FD52'     30   02AB   351          BSBW    MAC$DNUMBER                ;YES--GET IT
     50   0000'CF      B0   02AE   352          MOVW    W^MAC$GL_VALUE,R0          ;GET THE RESULT
     51   01    50     78   02B3   353          ASHL    R0,#1,R1                   ;GET 2**(VALUE)
                33     11   02B7   354          BRB     70$                        ;FINISH UP
                            02B9   355 ;
                            02B9   356 ; SYMBOLIC ALIGNMENT
                            02B9   357 ;
             FD44'     30   02B9   358 30$:     BSBW    MAC$SYMSCNUP               ;SCAN THE SYMBOL
             18 50     E9   02BC   359          BLBC    R0,50$                     ;BRANCH IF NO SYMBOL SCANNED
  55    00000000'EF    9E   02BF   360          MOVAB   L^PSC$G_OPTIONS,R5         ;POINT TO PSECT OPTIONS
             FD37'     30   02C6   361          BSBW    MAC$SRC_LIST               ;LOOK IT UP
             0B 50     E9   02C9   362          BLBC    R0,50$                     ;BRANCH IF NOT FOUND
       50    05 A1     D0   02CC   363          MOVL    SYM$L_VAL(R1),R0           ;GET THE BITS FOR SYMBOL
     50    03FF 8F     B3   02D0   364          BITW    #PSC$M_ALLOPTNS,R0         ; Is this an alignment?
                0C     13   02D5   365          BEQL    60$                        ;IF EQL YES
                            02D7   366 ;
                            02D7   367 ; INVALID ALIGNMENT
                            02D7   368 ;
                            02D7   369 50$:     $MAC_ERR INVALIGN                  ; Get message code
             FD21'     30   02DC   370          BSBW    MAC$ERRORLN                ;ISSUE TO PASS 2
                50     7C   02DF   371          CLRQ    R0                         ;DO NO ALIGNING
                09     11   02E1   372          BRB     70$                        ;...
                            02E3   373 ;
                            02E3   374 ; FINISH SYMBOL ALIGNMENT
                            02E3   375 ;
                0A     EF   02E3   376 60$:     EXTZV   #PSC$V_ALIGNMENT,-         ;GET EXPONENT VALUE FOR KEYWORD
     50   50    04          02E5   377                  #PSC$S_ALIGNMENT,R0,R0
     51   01    50     78   02E8   378          ASHL    R0,#1,R1                   ;CALCULATE ALIGNMENT FACTOR
     52   0000'CF      D0   02EC   379 70$:     MOVL    W^MAC$GL_PSECTPTR,R2       ;POINT TO CURRENT PSECT
                0A     EF   02F1   380          EXTZV   #PSC$V_ALIGNMENT,-         ;GET PSECT ALIGNMENT
                04          02F3   381                  #PSC$S_ALIGNMENT,-
        53    0D A2         02F4   382                  PSC$W_OPTIONS(R2),R3
     54   01    53     78   02F7   383          ASHL    R3,#1,R4                   ;CALCULATE PSECT ALIGNMENT
          54    51     D1   02FB   384          CMPL    R1,R4                      ;ALIGNMENT TOO BIG
                0B     1B   02FE   385          BLEQU   80$                        ;IF LEQU NO
                            0300   386          $MAC_ERR ALIGNXCEED                ; Yes--set code
             FCF8'     30   0305   387          BSBW    MAC$ERRORLN                ;ISSUE TO PASS 2
          51    54     D0   0308   388          MOVL    R4,R1                      ;USE PSECT ALIGNMENT
   0000'CF      51     D0   030B   389 80$:     MOVL    R1,W^MAC$GL_EXPOPVL1       ;STORE VALUE FOR ALIGN1/ALIGN2
```

```
 05  0310    390          RSB
```

```
                        0311  ;++
                        0311  ; FUNCTIONAL DESCRIPTION:
                        0311  ;
                        0311  ;       ALIGN1/ALIGN2 ARE CALLED TO FINISH PROCESSING THE .ALIGN
                        0311  ;       DIRECTIVE.  IF NECESSARY, THE PC IS ADJUSTED AND, IF THIS
                        0311  ;       IS ALIGN2, THEN THE FILL IS EMITTED TO PASS 2 ALSO.
                        0311  ;
                        0311  ;--
                        0311
                        0311  ALIGN1::                                        ;DIRECTIVE = ALIGN_HEAD
       06 6B    04  E5  0311          BBCC    #FLG$V_DATRPT,(R11),ALIGN_COM ;FLAG NO FILL EXPRESSION
                04  11  0315          BRB     ALIGN_COM                     ;(YOU NEVER CAN TELL...)
                        0317
                        0317  ALIGN2::                                      ;DIRECTIVE = ALIGN_HEAD EXPR
                        0317                                                ;DIRECTIVE = ALIGN_HEAD DCOMMA EXPR
       00 6B    04  E3  0317          BBCS    #FLG$V_DATRPT,(R11),ALIGN_COM ;FLAG FILL EXPRESSION PRESENT
                        031B  ALIGN_COM:
          0000'CF   D5  031B          TSTL    W^MAC$GL_ABSFLAG              ;ABSOLUTE EXPRESSION?
                08  13  031F          BEQL    10$                           ;IF EQL YES
                        0321          $MAC_ERR INVALIGN                     ; No--get error code
             FCD7'  30  0326          BSBW    MAC$ERRORPT                   ;ISSUE ERROR TO PASS 2
    52    0000'CF   D0  0329  10$:    MOVL    W^MAC$GL_EXPOPVL1,R2          ;GET THE ALIGNMENT FACTOR
                46  13  032E          BEQL    40$                           ;IF EQL NONE
          53    52  D0  0330          MOVL    R2,R3                         ;COPY IT
                52  D7  0333          DECL    R2                            ;
          52    52  D2  0335          MCOML   R2,R2                         ;AND COMPLEMENT
    51    0000'CF   D0  0338          MOVL    W^MAC$GL_PC,R1                ;GET CURRENT PC
          51    52  CA  033D          BICL2   R2,R1                         ;
                34  13  0340          BEQL    40$                           ;IF EQL NO ADJUSTMENT NEEDED
          53    51  C2  0342          SUBL2   R1,R3                         ;
                53  DD  0345          PUSHL   R3                            ;
       0A 6B    04  E4  0347          BBSC    #FLG$V_DATRPT,(R11),20$ ;IS THERE A FILL?
                        034B          $INTOUT_LW INT$_AUGPC,R3              ;NO--AUGMENT PC
                19  11  0353          BRB     30$                           ;AND EXIT
                        0355  ;
                        0355  ; THERE IS A FILL
                        0355  ;
                        0355  20$:    $INTOUT_LW INT$_STKL,<W^MAC$AL_VALSTACK[R7]> ;STACK THE FILL EXPR
                        0360          $INTOUT_LW INT$_STKL,R3               ;STACK THE FILL COUNT
                        0368          $INTOUT_X INT$_STRB                   ;STORE REPEATED BYTE
             53 8ED0    036E  30$:    POPL    R3                            ;GET THE PC AUGMENTATION
                        0371          $INC_PC R3                            ;ADJUST PC IN PASS 1 ALSO
                    05  0376  40$:    RSB
                        0377
                        0377          .END
```

```
$COUNT         = 0000003B                 FLG$M_MEBLST   = 00001000        FLG$V_NULCHR   = 00000032
ALIGN            00000296 RG    03        FLG$M_MOREARG  = 00002000        FLG$V_OBJXST   = 00000015
ALIGN1           00000311 RG    03        FLG$M_MOREINP  = 00000008        FLG$V_OPNDCHK  = 00000028
ALIGN2           00000317 RG    03        FLG$M_NEWPND   = 00000400        FLG$V_OPRND    = 0000000D
ALIGN_COM        0000031B R     03        FLG$M_NOREF    = 01000000        FLG$V_OPTVFLIDX= 0000002C
ARG$K_SIZE     = 000003E8                 FLG$M_NTYPEPC  = 00000020        FLG$V_ORDLST   = 00000011
AUD$K_SIZE     = 00000010                 FLG$M_NULCHR   = 00040000        FLG$V_P2       = 0000000E
BLNK           = 00000020                 FLG$M_OBJXST   = 00200000        FLG$V_RPTIRP   = 0000001C
CHR$M_COMMA_CR = 00000020                 FLG$M_OPNDCHK  = 00000100        FLG$V_SEQFIL   = 00000019
CHR$M_ILL_CHR  = 00000040                 FLG$M_OPRND    = 00002000        FLG$V_SKAN     = 0000000F
CHR$M_NUM_BER  = 00000010                 FLG$M_OPTVFLIDX= 00001000        FLG$V_SPECOP   = 00000022
CHR$M_SPA_MSK  = 00000001                 FLG$M_ORDLST   = 00020000        FLG$V_SPLALL   = 0000001A
CHR$M_SYM_CH1  = 00000008                 FLG$M_P2       = 00004000        FLG$V_STOIMF   = 00000012
CHR$M_SYM_CHR  = 00000004                 FLG$M_RPTIRP   = 10000000        FLG$V_SYM2COL  = 0000002A
CHR$M_SYM_DLM  = 00000002                 FLG$M_SEQFIL   = 02000000        FLG$V_TOCFLG   = 00000013
CHR$V_COMMA_CR = 00000005                 FLG$M_SKAN     = 00008000        FLG$V_UPAFLG   = 00000024
CHR$V_CVTLWC   = 00000061                 FLG$M_SPECOP   = 00000004        FLG$V_UPDFIL   = 00000027
CHR$V_ILL_CHR  = 00000006                 FLG$M_SPLALL   = 04000000        FLG$V_UPMARG   = 00000026
CHR$V_NOCVT    = 0000007F                 FLG$M_STOIMF   = 00040000        FLG$V_XCRF     = 0000001F
CHR$V_NUM_BER  = 00000004                 FLG$M_SYM2COL  = 00000400        HASHSZ         = 0000007F
CHR$V_SPA_MSK  = 00000000                 FLG$M_TOCFLG   = 00080000        HYPHEN         = 0000002D
CHR$V_SYM_CH1  = 00000003                 FLG$M_UPAFLG   = 00000010        INP$K_BUFSIZ   = 000003E8
CHR$V_SYM_CHR  = 00000002                 FLG$M_UPDFIL   = 00000080        INT$K_BUFSIZ   = 000013F4
CHR$V_SYM_DLM  = 00000001                 FLG$M_UPMARG   = 00000040        INT$K_BUFWRN   = 00001390
CNT            = 00000001                 FLG$M_XCRF     = 80000000        INT$_ADD       = 00000001
CR             = 0000000D                 FLG$V_ALLCHR   = 00000000        INT$_AND       = 00000002
CRESTO           00000224 RG    03        FLG$V_BOL      = 00000001        INT$_ASH       = 00000003
CSAVE            000001BD RG    03        FLG$V_CHKLPND  = 00000014        INT$_ASN       = 0000000C
ENB$G_LOCALSYMB  ********   X   03        FLG$V_COMPEXPR = 00000002        INT$_AUGPC     = 0000000D
ERR            = 00000000                 FLG$V_CONT     = 00000003        INT$_BDST      = 0000000E
FF             = 0000000C                 FLG$V_CRF      = 0000001E        INT$_CHKL      = 0000000F
FLG$M_ALLCHR   = 00000001                 FLG$V_CRSEEN   = 00000020        INT$_DIV       = 00000004
FLG$M_BOL      = 00000002                 FLG$V_DATRPT   = 00000004        INT$_END       = 00000010
FLG$M_CHKLPND  = 00100000                 FLG$V_DBGOUT   = 0000002E        INT$_EPT       = 00000011
FLG$M_COMPEXPR = 00000004                 FLG$V_DLIMSTR  = 0000002F        INT$_ERR       = 00000012
FLG$M_CONT     = 00000008                 FLG$V_ENDMCH   = 00000005        INT$_ETX       = 00000013
FLG$M_CRF      = 40000000                 FLG$V_EVALEXPR = 00000006        INT$_FNEWL     = 00000014
FLG$M_CRSEEN   = 00000001                 FLG$V_EXPOPT   = 00000007        INT$_ILG       = 00000000
FLG$M_DATRPT   = 00000010                 FLG$V_EXTERR   = 00000030        INT$_INFO      = 0000003A
FLG$M_DBGOUT   = 00004000                 FLG$V_EXTWRN   = 00000031        INT$_LGLAB     = 00000015
FLG$M_DLIMSTR  = 00008000                 FLG$V_FIRSTLN  = 00000029        INT$_MACL      = 00000016
FLG$M_ENDMCH   = 00000020                 FLG$V_IFSTAT   = 00000017        INT$_MUL       = 00000005
FLG$M_EVALEXPR = 00000040                 FLG$V_IIF      = 00000016        INT$_NEG       = 00000006
FLG$M_EXPOPT   = 00000080                 FLG$V_INSERT   = 00000008        INT$_NEWL      = 00000017
FLG$M_EXTERR   = 00010000                 FLG$V_IRPC     = 0000001D        INT$_NEWP      = 00000018
FLG$M_EXTWRN   = 00020000                 FLG$V_LEXOP    = 00000021        INT$_NOT       = 00000007
FLG$M_FIRSTLN  = 00000200                 FLG$V_LSTXST   = 00000009        INT$_OP        = 00000019
FLG$M_IFSTAT   = 00800000                 FLG$V_MAC2COL  = 0000002B        INT$_OR        = 00000008
FLG$M_IIF      = 00400000                 FLG$V_MACL     = 0000000B        INT$_PRIL      = 0000001A
FLG$M_INSERT   = 00000100                 FLG$V_MACLTB   = 0000001B        INT$_PRT       = 0000001B
FLG$M_IRPC     = 20000000                 FLG$V_MACTXT   = 00000010        INT$_PSECT     = 0000001C
FLG$M_LEXOP    = 00000002                 FLG$V_MEBLST   = 0000000C        INT$_REDEF     = 0000001D
FLG$M_LSTXST   = 00000200                 FLG$V_MOREARG  = 0000002D        INT$_REF       = 0000001E
FLG$M_MAC2COL  = 00000800                 FLG$V_MOREINP  = 00000023        INT$_REST      = 0000001F
FLG$M_MACL     = 00000800                 FLG$V_NEWPND   = 0000000A        INT$_SAME      = 00000009
FLG$M_MACLTB   = 08000000                 FLG$V_NOREF    = 00000018        INT$_SAVE      = 00000020
FLG$M_MACTXT   = 00010000                 FLG$V_NTYPEPC  = 00000025        INT$_SBTTL     = 00000021
```

```
INT$_SETFLAG   = 00000022                 MAC$SET_NEW_LSB   ********       X   03    PSC$M_WORD     = 00004400
INT$_SETLONG   = 00000023                 MAC$SET_PC        ********       X   03    PSC$M_WRT      = 00000180
INT$_SPIC      = 00000024                 MAC$SKIPSP        ********       X   03    PSC$S_ALIGNMENT= 00000004
INT$_SPID      = 00000025                 MAC$SKP_OPR       ********       X   03    PSC$V_ALIGNFLG = 0000000E
INT$_STIB      = 00000026                 MAC$SRC_LIST      ********       X   03    PSC$V_ALIGNMENT= 0000000A
INT$_STIL      = 00000028                 MAC$SYMSCNUP      ********       X   03    PSC$V_EXE      = 00000006
INT$_STIW      = 00000027                 MAC$_ALIGNXCEED= 007D900A                  PSC$V_GBL      = 00000004
INT$_STKEPT    = 00000029                 MAC$_DIRSYNX   = 007D906A                  PSC$V_LIB      = 00000001
INT$_STKG      = 0000002A                 MAC$_INVALIGN  = 007D912A                  PSC$V_OVR      = 00000002
INT$_STKL      = 0000002B                 MAC$_NOTPSECOPT= 007D919A                  PSC$V_PIC      = 00000000
INT$_STKPC     = 0000002C                 MAC$_PSECBUFOVF= 007D91BA                  PSC$V_RD       = 00000007
INT$_STKS      = 0000002D                 MAC$_PSECBUFUND= 007D91C2                  PSC$V_REL      = 00000003
INT$_STOB      = 00000034                 MAC$_PSECOPCNFL= 007D91B2                  PSC$V_SHR      = 00000005
INT$_STOL      = 0000002E                 MAC$_TOOMNYPSEC= 007D920A                  PSC$V_VEC      = 00000009
INT$_STOW      = 00000035                 MAC_SUBSYS     = 0000007D                  PSC$V_WRT      = 00000008
INT$_STRB      = 0000002F                 OBJ$K_BUFSIZ   = 00000200                  PSC$W_FLAG       00000009
INT$_STRL      = 00000031                 OPF$M_LASTOPR  = 00002000                  PSC$W_OPTIONS    0000000D
INT$_STRSB     = 00000032                 OPF$M_OPTEXP   = 00001000                  PSECT            00000000 RG        03
INT$_STRSW     = 00000033                 OPF$V_LASTOPR  = 0000000D                  PSECT$BLANK      ********       X   03
INT$_STRW      = 00000030                 OPF$V_OPTEXP   = 0000000C                  RDX$V_BINARY   = 00000000
INT$_STSB      = 00000036                 OPTION_CONFLICT  000001B5 R          03    RDX$V_DECIMAL  = 00000002
INT$_STSW      = 00000037                 OPTION_SCAN      000000BF R          03    RDX$V_DOUBLE   = 00000005
INT$_SUB       = 0000000A                 PSC$B_NAME       00000004                  RDX$V_FLOAT    = 00000004
INT$_SUME      = 00000039                 PSC$B_SEG        0000000C                  RDX$V_GFLOAT   = 00000006
INT$_WRN       = 00000038                 PSC$B_UNUSED     0000000B                  RDX$V_HEX      = 00000003
INT$_XOR       = 0000000B                 PSC$G_OPTIONS    ********       X   03    RDX$V_HFLOAT   = 00000007
LST$K_BUFSIZ   = 00000086                 PSC$K_BLKSIZ     00000013                  RDX$V_OCTAL    = 00000001
LST$K_L_P_PAGE = 0000003C                 PSC$K_NO_OPTNS = 0000000A                  REG$_PC        = 0000000F
LST$K_TITLE_SIZ= 00000028                 PSC$L_CURLOC     0000000F                  SEMI           = 0000003B
MAC$AB_CMSK_TAB  ********       X   03    PSC$L_LINK       00000000                  STB$K_PG_MISS  = 0000000A
MAC$AB_PSC_SBF   ********       X   03    PSC$L_MAXLGTH    00000005                  SYM$B_NAME       00000004
MAC$AB_TMPSYM    ********       X   03    PSC$M_ABS      = FFFFFFF7                  SYM$B_SEG        0000000C
MAC$ALL_2_PAGES  ********       X   03    PSC$M_ALIGNFLG = 00004000                  SYM$B_TOKEN      0000000B
MAC$AL_PSC_SLB   ********       X   03    PSC$M_ALLOPTNS = 000003FF                  SYM$K_BLKSIZ     0000000D
MAC$AL_VALSTACK  ********       X   03    PSC$M_BYTE     = 00004000                  SYM$K_MAXLEN   = 0000001F
MAC$DNUMBER      ********       X   03    PSC$M_CON      = FFFFFFFB                  SYM$K_TWOCOL   = 00000010
MAC$ERRORLN      ********       X   03    PSC$M_DEFAULT  = 000001C8                  SYM$L_LINK       00000000
MAC$ERRORPT      ********       X   03    PSC$M_EXE      = 000000C0                  SYM$L_VAL        00000005
MAC$ERR_INTERN   ********       X   03    PSC$M_GBL      = 00000010                  SYM$M_ABS      = 00000010
MAC$GETCHR       ********       X   03    PSC$M_LCL      = FFFFFFEF                  SYM$M_ASN      = 00000100
MAC$GL_ABSFLAG   ********       X   03    PSC$M_LIB      = 00000002                  SYM$M_CRFO     = 00002000
MAC$GL_EXPOPVL1  ********       X   03    PSC$M_LONG     = 00004800                  SYM$M_DEBUG    = 00000020
MAC$GL_LSB       ********       X   03    PSC$M_NOEXE    = FFFFFFBF                  SYM$M_DEF      = 00000001
MAC$GL_PC        ********       X   03    PSC$M_NOPIC    = FFFFFFFE                  SYM$M_DELMAC   = 00000200
MAC$GL_PSC_BLKP  ********       X   03    PSC$M_NORD     = FFFFFF7F                  SYM$M_EPT      = 00000200
MAC$GL_PSC_LIST  ********       X   03    PSC$M_NOSHR    = FFFFFFDF                  SYM$M_EXTRN    = 00000008
MAC$GL_PSC_MAX   ********       X   03    PSC$M_NOVEC    = FFFFFDFF                  SYM$M_GLOBL    = 00000004
MAC$GL_PSC_SBP   ********       X   03    PSC$M_NOWRT    = FFFFFEFF                  SYM$M_LOCAL    = 00000040
MAC$GL_PSECT     ********       X   03    PSC$M_OVR      = 00000004                  SYM$M_ODBG     = 00000400
MAC$GL_PSECTPTR  ********       X   03    PSC$M_PAGE     = 00006400                  SYM$M_REF      = 00000080
MAC$GL_VALUE     ********       X   03    PSC$M_PIC      = 00000001                  SYM$M_RELPSECT = 00000800
MAC$G_LSBNAM     ********       X   03    PSC$M_QUAD     = 00004C00                  SYM$M_SUPR     = 00004000
MAC$INTOUT_1_LW  ********       X   03    PSC$M_RD       = 00000080                  SYM$M_WEAK     = 00000002
MAC$INTOUT_X     ********       X   03    PSC$M_REL      = 00000008                  SYM$M_XCRF     = 00001000
MAC$LAST_CHANCE  ********       X   03    PSC$M_SHR      = 00000020                  SYM$V_ABS      = 00000004
MAC$PSC_OPT_SCN  000000A3 R          03    PSC$M_USR      = FFFFFFFD                  SYM$V_ASN      = 00000008
MAC$PSC_RES_ERR  0000028E RG         03    PSC$M_VEC      = 00000200                  SYM$V_CRFO     = 0000000D
```

```
SYM$V_DEBUG     = 00000005
SYM$V_DEF       = 00000000
SYM$V_DELMAC    = 00000009
SYM$V_EPT       = 00000009
SYM$V_EXTRN     = 00000003
SYM$V_GLOBL     = 00000002
SYM$V_LOCAL     = 00000006
SYM$V_ODBG      = 0000000A
SYM$V_REF       = 00000007
SYM$V_RELPSECT  = 0000000B
SYM$V_SUPR      = 0000000E
SYM$V_WEAK      = 00000001
SYM$V_XCRF      = 0000000C
SYM$W_FLAG        00000009
TAB             = 00000009
X1              = 00000400
X2              = 0000000F

                                                  +----------------+
                                                  ! Psect synopsis !
                                                  +----------------+

PSECT name                     Allocation           PSECT No.  Attributes
----------                     ----------           ---------  ----------
.  ABS  .                      00000000  (    0.)   00 (  0.)  NOPIC     USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
. BLANK .                      00000000  (    0.)   01 (  1.)  NOPIC     USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$ABS$                          00000013  (   19.)   02 (  2.)  NOPIC     USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
MAC$RO_CODE_P1                 00000377  (  887.)   03 (  3.)  NOPIC     USR    CON    REL    GBL NOSHR   EXE   RD   NOWRT NOVEC LONG

                                               +------------------------+
                                               ! Performance indicators !
                                               +------------------------+

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                   35     00:00:00.04    00:00:01.47
Command processing              128     00:00:00.37    00:00:04.68
Pass 1                          208     00:00:03.42    00:00:14.44
Symbol table sort                 0     00:00:00.41    00:00:01.64
Pass 2                           95     00:00:00.91    00:00:04.17
Symbol table output              29     00:00:00.15    00:00:00.23
Psect synopsis output             2     00:00:00.01    00:00:00.02
Cross-reference output            0     00:00:00.00    00:00:00.00
Assembler run totals            499     00:00:05.32    00:00:26.65

The working set limit was 1350 pages.
32128 bytes (63 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 431 non-local and 45 local symbols.
436 source lines were read in Pass 1, producing 19 object records in Pass 2.
14 pages of virtual memory were used to define 13 macros.
```

```
                                             +---------------------------+
                                             ! Macro library statistics !
                                             +---------------------------+

Macro library name                           Macros defined
------------------                           --------------
_$255$DUA28:[MACRO.OBJ]MACRO.MLB;1                 11
_$255$DUA28:[SYSLIB]STARLET.MLB;2                   3
TOTALS (all libraries)                             14

505 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:APSECT/OBJ=OBJ$:APSECT MSRC$:APSECT/UPDATE=(ENH$:APSECT)+LIB$:MACRO/LIB
```

0224 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY