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
```

FILEID**P2DRVR

F 4

MA
VO

```
PPPPPPPP      222222    DDDDDDDD    RRRRRRRR    VV      VV  RRRRRRRR
PPPPPPPP      222222    DDDDDDDD    RRRRRRRR    VV      VV  RRRRRRRR
PP      PP  22      22  DD      DD  RR      RR  VV      VV  RR      RR
PP      PP  22      22  DD      DD  RR      RR  VV      VV  RR      RR
PP      PP          22  DD      DD  RR      RR  VV      VV  RR      RR
PP      PP          22  DD      DD  RR      RR  VV      VV  RR      RR
PPPPPPPP          22    DD      DD  RRRRRRRR    VV      VV  RRRRRRRR
PPPPPPPP          22    DD      DD  RRRRRRRR    VV      VV  RRRRRRRR
PP              22      DD      DD  RR  RR      VV      VV  RR  RR
PP              22      DD      DD  RR  RR      VV      VV  RR  RR
PP            22        DD      DD  RR    RR      VV  VV      RR    RR        ....
PP            22        DD      DD  RR    RR      VV  VV      RR    RR        ....
PP          2222222222  DDDDDDDD    RR      RR      VV      RR      RR      ....
PP          2222222222  DDDDDDDD    RR      RR      VV      RR      RR      ....

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
```

42

Table of contents

```
0000            .TITLE  MAC$P2DRVR PASS 2 DRIVER MODULE
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    ;       V03-001 MTR0033         Mike Rhodes     25-Apr-1983
0000    ;               Fix link truncation errors.
0000    ;
0000    ;       V02.08  PCG0002         Peter George    16-Apr-1981
0000    ;               Filter out abs psect code.
0000    ;
0000    ;       V01.07  RN0005          R. Newland      13-Aug-1979
0000    ;               Variable symbol storage and remove .ALIGN LONG statements
0000    ;
0000    ;       V01.07  RN0022          R. Newland      31-Oct-1979
0000    ;               Translate SYS$LP_LINES to set lines/page
0000    ;
0000    ;       V01.06  RN0002          R. Newland      01-Feb-1979
0000    ;               Changes for Source Update Merge
0000    ; 01    -
```

```
                    0000    58 ;--
```

```
               0000         .SBTTL  DECLARATIONS
               0000  ;
               0000  ; INCLUDE FILES:
               0000  ;
               0000
               0000  ;
               0000  ; MACROS:
               0000  ;
               0000
               0000         $RABDEF                         ;DEFINE RAB OFFSETS
               0000         $MAC_GENVALDEF                  ;DEFINE GENERAL VALUES
               0000         $MAC_SYMBLKDEF                  ;DEFINE SYMBOL BLOCK OFFSETS
               0000         $MAC_CTLFLGDEF                  ;DEFINE CONTROL FLAG OFFSETS
               0000
               0000  ;
               0000  ; LOCAL DATA
               0000  ;
               0000
           00000000         .PSECT  MAC$RO_DATA,NOEXE,NOWRT,GBL,LONG
               0000
               0000  P2$DISPATCH::
               0000         $MAC_INTCODDEF  DISPATCHTABLE
  0000003B     00EC  P2$K_MAXCOD=$COUNT                     ;MAXIMUM PASS 2 CODE + 1
```

```
                                00EC     84         .SBTTL  MAC$P2DRVR  PASS 2 DRIVER
                                00EC     85
                                00EC     86 ;++
                                00EC     87 ; FUNCTIONAL DESCRIPTION:
                                00EC     88 ;
                                00EC     89 ;       THIS ROUTINE OUTPUTS THE MODULE HEADER TO THE OBJECT FILE,
                                00EC     90 ;       INITIALIZES TO READ THE INTERMEDIATE 'FILE', AND THEN
                                00EC     91 ;       PROCESSES THE INTERMEDIATE FILE.
                                00EC     92 ;
                                00EC     93 ;--
                                00EC     94
                            00000000     95         .PSECT  MAC$RO_CODE_P2,NOWRT,GBL,LONG
                                0000     96
                                0000     97 MAC$PASS2_DRIVR::
          5A    00000000'GF  9E 0000     98         MOVAB   G^MAC$AB_OBJBUF,R10     ;R10 POINTS TO OBJECT BUFFER DURING PASS 2
                00000000'GF  D4 0007     99         CLRL    G^MAC$GL_PC             ;PC STARTS AT 0
                         57  D4 000D    100         CLRL    R7                      ;INITIALIZE STACK POINTER
                       031F  30 000F    101         BSBW    MAC$WRT_LST_INI         ;INITIALIZE LISTING BUFFER
                   06 6B  29 E4 0012    102         BBSC    #FLG$V_FIRSTLN,(R11),10$ ; Branch if immediate error
                      FFE7'  30 0016    103         BSBW    MAC$GETLIN              ;START WITH FIRST LINE OF FILE
                      FFE4'  30 0019    104         BSBW    MAC$FORM_LINENO         ; Form line number and audit trail
                                001C    105 10$:
 00000000'GF    00000000'GF  9E 001C    106         MOVAB   G^PSECT$MAIN,G^MAC$GL_PSECTPTR ;START IN BLANK PSECT
                                0027    107 ;
                                0027    108 ; OUTPUT MODULE HEADER INFORMATION
                                0027    109 ;
                      FFD6'  30 0027    110         BSBW    MAC$OBJHDROUT           ;OUTPUT MODULE HEADER INFO
                                002A    111 ;
                                002A    112 ; RESET THE INITIAL SETTINGS FOR THE ENABLE/DISABLE AND LIST/NLIST
                                002A    113 ; OPTIONS
                                002A    114 ;
          55    00000000'EF  9E 002A    115         MOVAB   L^LST$G_DIRLIST,R5      ;POINT TO THE LIST
            05 A5     08 A5  98 0031    116 2$:     CVTBL   SYM$B_TOKEN(R5),SYM$L_VAL(R5) ;RESET TO THE INITIAL SETTING
                   55     65 D0 0036    117         MOVL    (R5),R5                 ;LINK TO NEXT
                         F6  12 0039    118         BNEQ    2$                      ;LOOP FOR ALL
          55    00000000'EF  9E 003B    119         MOVAB   L^ENB$G_OPTIONS,R5      ;POINT TO ENABLE OPTIONS LIST
            05 A5     08 A5  98 0042    120 3$:     CVTBL   SYM$B_TOKEN(R5),SYM$L_VAL(R5) ;RESET
                   55     65 D0 0047    121         MOVL    (R5),R5
                         F6  12 004A    122         BNEQ    3$                      ;LOOP FOR ALL
                                004C    123 ;
                                004C    124 ; SET UP TO READ INTERMEDIATE 'FILE' FROM VIRTUAL MEMORY
                                004C    125 ;
          59    00000000'GF  D0 004C    126         MOVL    G^MAC$GL_INTQUE,R9      ;POINT TO BEGINNING OF FIRST BLOCK
              58      0C A9  9E 0053    127         MOVAB   12(R9),R8               ;R8 POINTS TO FIRST COMMAND
                         57  D4 0057    128         CLRL    R7                      ;INIT VALUE STACK POINTER
          00000000'GF    5E  D0 0059    129         MOVL    SP,G^MAC$GL_SAVE_SP     ;SAVE STACK POINTER
          7E    58    08 A9  C1 0060    130         ADDL3   8(R9),R8,-(SP)          ;FIGURE END OF BLOCK ONTO STACK
                                0065    131 ;
                                0065    132 ; AT THIS POINT:
                                0065    133 ;
                                0065    134 ;       R10     POINTER INTO OBJECT BUFFER
                                0065    135 ;       R9      POINTER TO CURRENT INTERMEDIATE 'FILE' BLOCK
                                0065    136 ;       R8      POINTER INTO INTERMEDIATE FILE (INCH-ALONG)
                                0065    137 ;       R7      VALUE STACK POINTER
                                0065    138 ;
                                0065    139 PASS_2_LOOP:
                   6E     58 D1 0065    140         CMPL    R8,(SP)                 ;ARE WE AT THE END OF A BUFFER?
```

```
                 0E   1F  0068              BLSSU   10$                          ;IF LSSU NO
             59  69   D0  006A              MOVL    (R9),R9                      ;YES--CHAIN TO NEXT BUFFER
                 49   13  006D              BEQL    MAC$EARLY_END                ;IF EQL THEN EARLY ENDING
         58  0C  A9   9E  006F              MOVAB   12(R9),R8                    ;POINT TO FIRST ACTION
     6E  58  08  A9   C1  0073              ADDL3   8(R9),R8,(SP)                ;FIGURE END OF NEW BUFFER
             56  88   9A  0078      10$:    MOVZBL  (R8)+,R6                     ;GET LENGTH OF FRAME
             56  02   C2  007B              SUBL2   #2,R6                        ;COUNT LENGTH AND COMMAND BYTES
             50  88   9A  007E              MOVZBL  (R8)+,R0                     ;GET COMMAND
                 05   15  0081              BLEQ    20$                          ;IF LE THEN SURELY ILLEGAL
             3B  50   91  0083              CMPB    R0,#P2$K_MAXCOD              ;ENSURE LEGAL
                 02   19  0086              BLSS    30$                          ;IF LSS THEN LEGAL
                 50   D4  0088      20$:    CLRL    R0                           ;0 GOES TO ILLEGAL COMMAND ROUTINE
                          008A      ;
                          008A      ; CALL PASS 2 ACTION ROUTINE WITH:
                          008A      ;
                          008A      ;               R10     POINTER INTO OBJECT BUFFER
                          008A      ;               R9      RESERVED
                          008A      ;               R8      POINTER TO INTERMEDIATE FILE
                          008A      ;               R7      VALUE STACK POINTER
                          008A      ;               R6      LENGTH OF COMMAND IN BYTES
                          008A      ;               R5      POINTER TO TOP OF VALUE STACK (MAC$AL_VALSTACK[R7])
                          008A      ;
                          008A      ;               REGISTERS R8, R6, R5, AND R4-R0 MAY BE DESTROYED
                          008A      ;
        FD 8F    56   91  008A      30$:    CMPB    R6,#<-1-2>                   ;SPECIAL MACRO LINE (-1 FLAG -2 BYTES)?
                 06   12  008E              BNEQ    40$                          ;IF NEQ NO
             56  68   3C  0090              MOVZWL  (R8),R6                      ;YES--GET LENGTH OF TEXT
             56  02   C0  0093              ADDL2   #2,R6                        ;COUNT THE LENGTH WORD
        7E   58  56   C1  0096      40$:    ADDL3   R6,R8,-(SP)                  ;FIGURE START OF NEXT COMMAND
55    00000000'GF47   DE  009A              MOVAL   G^MAC$AL_VALSTACK[R7],R5 ;POINT TO TOP OF VALUE STACK
      00000000'EF40   DD  00A2              PUSHL   L^P2$DISPATCH[R0]            ;GET ACTION ROUTINE ADDRESS
                 9E   16  00A9              JSB     @(SP)+                       ;DISPATCH TO ACTION ROUTINE
             58  8E   D0  00AB              MOVL    (SP)+,R8                     ;SET POINTER TO NEXT COMMAND
                 B5   11  00AE              BRB     PASS_2_LOOP
                          00B0      ;
                          00B0      ; WHEN THE .END DIRECTIVE IS ENCOUNTERED, CONTROL WILL TRANSFER
                          00B0      ; TO HERE.
                          00B0      ;
                          00B0      MAC$PASS_2_END::
  5E    00000000'GF   D0  00B0              MOVL    G^MAC$GL_SAVE_SP,SP          ;RESET THE STACK POINTER
                      05  00B7              RSB
                          00B8
                          00B8      ;
                          00B8      ; WE REACHED THE END OF THE INTERMEDIATE FILE BEFORE WE SHOULD HAVE??!!!
                          00B8      ;
                          00B8      MAC$EARLY_END::
  00000000'GF    00   FB  00B8              CALLS   #0,G^MAC$ERR_INTERN          ;REPORT INTERNAL ERROR
              FF3E'   31  00BF              BRW     P2$END                       ;AND GO FINISH PASS 2
                          00C2
                          00C2      ;
                          00C2      ; ILLEGAL COMMAND ENCOUNTERED IN INTERMEDIATE BUFFER
                          00C2      ;
                          00C2      P2$ILG::
  00000000'GF    00   FB  00C2              CALLS   #0,G^MAC$ERR_INTERN          ;REPORT INTERNAL ERROR
              FF34'   31  00C9              BRW     P2$END                       ;GO FINISH PASS 2
                          00CC      ;
                          00CC      ; STORE PIC CODE COMMAND NOT IMPLEMENTED (THIS SHOULD NEVAH HAPPEN SUH!)
```

```
                                 00CC     198 ;
                                 00CC     199 P2$SPIC::                                         ;(DO WE REALLY NEED THIS?)
    00000000'GF     00     FB    00CC     200             CALLS   #0,G^MAC$ERR_INTERN           ;REPORT INTERNAL ERROR
              FF2A'        31    00D3     201             BRW     P2$END                        ;GO FINISH PASS 2
```

```
                              00D6
                              00D6  ;++
                              00D6  ; FUNCTIONAL DESCRIPTION:
                              00D6  ;
                              00D6  ;       THIS ROUTINE EMITS A SYMBOL OR PSECT NAME TO THE OBJECT FILE.
                              00D6  ;
                              00D6  ; INPUTS:
                              00D6  ;
                              00D6  ;       R6      SYMBOL BLOCK ADDRESS
                              00D6  ;
                              00D6  ; OUTPUTS:
                              00D6  ;
                              00D6  ;       SYMBOL NAME AND SIZE OUTPUT TO OBJECT FILE
                              00D6  ;
                              00D6  ;--
                              00D6
                              00D6  MAC$SYMNAMOUT::
          0E 6B   2E    E0    00D6          BBS     #FLG$V_DBGOUT,(R11),5$      ; If debugger record
50    00000000'EF       D0    00DA          MOVL    MAC$GL_PSECTPTR,R0          ; or zero psect
                  05    13    00E1          BEQL    5$                          ; Then output record
      17 0D A0    03    E1    00E3          BBC     #PSC$V_REL, -               ; Else if abs psect, then
                              00E8                  PSC$W_OPTIONS(R0),20$       ; filter record out
        53    04 A6     9A    00E8  5$:     MOVZBL  SYM$B_NAME(R6),R3           ; Get offset to symbol count/name
      53    56    53    C3    00EC          SUBL3   R3,R6,R3                    ; and form its address
            54    83    90    00F0          MOVB    (R3)+,R4                    ; Get count and advance pointer to name
                              00F3          $OBJ_OUTBYT R4                      ;EMIT SIZE OF NAME
            8A    83    90    00F9  10$:    MOVB    (R3)+,(R10)+                ;STORE ONE CHARACTER OF NAME
              FA 54     F5    00FC          SOBGTR  R4,10$                      ;LOOP FOR WHOLE NAME
                  05          00FF  20$:    RSB
```

```
                            0100         .SBTTL  OBJECT FILE OUTPUT ROUTINES
                            0100
                            0100  ;++
                            0100  ; FUNCTIONAL DESCRIPTION:
                            0100  ;
                            0100  ;      OUTPUTS THE BYTE IN R0 TO THE STORE IMMEDIATE COMMAND
                            0100  ;      ALREADY IN PROGRESS.  IF THE FLAG 'FLG$M_STOIMF' IS
                            0100  ;      FALSE THEN A NEW STORE IMMEDIATE COMMAND IS STARTED WITH
                            0100  ;      THIS BYTE.  THE FIRST BYTE OF A STORE IMMEDIATE COMMAND
                            0100  ;      IS THE TWO'S COMPLEMENT BYTE COUNT OF THE STRING UP TO
                            0100  ;      -128. (^0200).
                            0100  ;
                            0100  ; INPUTS:
                            0100  ;
                            0100  ;      R0      BYTE TO STORE
                            0100  ;
                            0100  ;--
                            0100
                            0100  MAC$STOIM::
        0E 6B    2E    E0   0100         BBS     #FLG$V_DBGOUT,(R11),5$   ; If debugger record
   51   00000000'EF    D0   0104         MOVL    MAC$GL_PSECTPTR,R1       ; or zero psect
                 05    13   010B         BEQL    5$                       ; Then output record
     2D 0D A1    03    E1   010D         BBC     #PSC$V_REL, -            ; Else if abs psect, then
                            0112                 PSC$W_OPTIONS(R1),30$    ; filter record out
      0000'8F    5A    B1   0112  5$:    CMPW    R10,#MAC$AB_OBJWRN       ;PAST WARNING LIMIT?
                 02    1B   0117         BLEQU   10$                      ;IF LEQ NO
                 59    10   0119         BSBB    MAC$WRTOBJ               ;YES--OUTPUT BUFFER
        09 6B    12    E4   011B  10$:   BBSC    #FLG$V_STOIMF,(R11),20$ ;BRANCH IF STORE IMMEIDATE IN PROGRESS
    00000000'GF  5A    D0   011F         MOVL    R10,G^MAC$GL_STOIMPTR    ;NO--START ONE.
                 8A    94   0126         CLRB    (R10)+                   ;LEAVE ROOM FOR COUNT BYTE AND INIT IT
           8A    50    90   0128  20$:   MOVB    R0,(R10)+                ;STORE DATA BYTE
      00000000'FF      97   012B         DECB    @L^MAC$GL_STOIMPTR       ;COUNT THE BYTE
80 8F 00000000'FF      91   0131         CMPB    @L^MAC$GL_STOIMPTR,#^0200 ;DONE 128 YET?
                 04    1B   0139         BLEQU   30$                      ;IF LEQU YES
        00 6B    12    E2   013B         BBSS    #FLG$V_STOIMF,(R11),30$ ;NO--FLAG STORE IMMEDIATE IN PROGRESS
                       05   013F  30$:   RSB
```

```
                                0140  ;++
                                0140  ; FUNCTIONAL DESCRIPTION:
                                0140  ;
                                0140  ;       IF THE BUFFER POINTER IS PAST THE WARNING LIMIT, THE OBJECT
                                0140  ;       BUFFER IS WRITTEN OUT.  THE BYTE IN R0 IS THEN STORED IN THE
                                0140  ;       OBJECT FILE BUFFER.
                                0140  ;
                                0140  ; INPUTS:
                                0140  ;
                                0140  ;       R0      BYTE TO STORE
                                0140  ;
                                0140  ;--
                                0140
                                0140  MAC$CHKBYT::
           0000'8F     5A   B1  0140          CMPW    R10,#MAC$AB_OBJWRN      ;PAST THE WARNING LIMIT?
                       02   1B  0145          BLEQU   10$                     ;IF LEQU NO
                       2B   10  0147          BSBB    MAC$WRTOBJ              ;YES--OUTPUT RECORD
                       00   11  0149  10$:    BRB     MAC$OUTOBJ              ;OUTPUT THE BYTE
                                014B                                          ;AND RETURN
                                014B
                                014B  ;++
                                014B  ; FUNCTIONAL DESCRIPTION:
                                014B  ;
                                014B  ;       OUTPUT THE BYTE IN R0 TO THE OBJECT FILE
                                014B  ;
                                014B  ; INPUTS:
                                014B  ;
                                014B  ;       R0      BYTE TO OUTPUT
                                014B  ;
                                014B  ;--
                                014B
                                014B          .ENABLE LSB
                                014B
                                014B  MAC$OUTOBJ::
              0E 6B    2E   E0  014B          BBS     #FLG$V_DBGOUT,(R11),5$  ; If debugger record
      51   00000000'EF      D0  014F          MOVL    MAC$GL_PSECTPTR,R1      ; or zero psect
                       05   13  0156          BEQL    5$                      ; Then output record
           16 0D A1    03   E1  0158          BBC     #PSC$V_REL, -           ; Else if abs psect, then
                                015D                  PSC$W_OPTIONS(R1),20$   ; filter record out
00000000'GF   00000000'GF   91  015D  5$:     CMPB    G^MAC$GL_RECTYP,G^MAC$AB_OBJBUF ;SAME RECORD TYPE?
                       02   13  0168          BEQL    10$                     ;IF EQL YES
                       08   10  016A          BSBB    MAC$WRTOBJ              ;NO--WRITE OUT LAST RECORD
                                016C  10$:
                                016C  MAC$OUTOBJ_0::
                 8A    50   90  016C          MOVB    R0,(R10)+               ;STORE BYTE IN BUFFER
              00 6B    12   E5  016F          BBCC    #FLG$V_STOIMF,(R11),20$ ;NO LONGER STORING IMMEDIATE
                            05  0173  20$:    RSB
                                0174
                                0174          .DISABLE LSB
                            0174  ;++
                            0174  ; FUNCTIONAL DESCRIPTION:
                            0174  ;
                            0174  ;       WRITE OUT OBJECT CODE BUFFER TO OBJECT CODE FILE
                            0174  ;
                            0174  ; INPUTS:
                            0174  ;
                            0174  ;       R10     POINTS TO WHERE WE ARE IN BUFFER
                            0174  ;
                            0174  ; ***NO REGISTERS USED***
                            0174  ;
                            0174  ;--
                            0174
                            0174 MAC$WRTOBJ::
        5A   0001'8F   B1   0174          CMPW    #MAC$AB_OBJBUF+1,R10    ;IS BUFFER EMPTY?
                  37   1E   0179          BGEQU   10$                     ;IF GEQ YES
           33 6B  15   E1   017B          BBC     #FLG$V_OBJXST,(R11),10$ ;BRANCH IF NO OBJECT FILE
                  50   DD   017F          PUSHL   R0                      ;NO--SAVE R0
50      5A 00000000'8F C3   0181          SUBL3   #MAC$AB_OBJBUF,R10,R0   ;FIGURE LENGTH OF RECORD
   00000022'EF    50   B0   0189          MOVW    R0,L^MAC$OBJECT_RAB+RAB$W_RSZ ;SET RECORD LENGTH IN RAB
                            0190          $PUT    RAB=L^MAC$OBJECT_RAB,-  ;SEND RECORD TO OUTPUT FILE
                            0190                  ERR=G^MAC$ERR_PUT       ; ERROR ROUTINE
              03 50    E8   01A3          BLBS    R0,5$                   ;BRANCH IF SUCCESSFUL
               FE57'   30   01A6          BSBW    MAC$CLS_DEL_OBJ         ;CLOSE AND DELETE OBJECT FILE
          00000000'GF  D6   01A9 5$:      INCL    G^MAC$GL_OBJ_RCNT       ;COUNT OBJECT RECORD WRITTEN
                  50 8ED0   01AF          POPL    R0                      ;RESTORE R0
     5A   00000000'GF  9E   01B2 10$:     MOVAB   G^MAC$AB_OBJBUF,R10     ;RESET POINTER INTO BUFFER
     8A   00000000'GF  90   01B9          MOVB    G^MAC$GL_RECTYP,(R10)+  ;STORE RECORD TYPE IN BUFFER
           00 6B  12   E5   01C0          BBCC    #FLG$V_STOIMF,(R11),20$ ;STOP ANY STORE IMMEDIATE IN PROGRESS
                  05        01C4 20$:     RSB
```

```
                           01C5          .SBTTL  LISTING FILE ROUTINES
                           01C5
                           01C5     ;++
                           01C5     ; FUNCTIONAL DESCRIPTION:
                           01C5     ;
                           01C5     ;          POP THE TOP LONGWORD OFF THE VALUE STACK AND LIST THE NUMBER
                           01C5     ;          OF BYTES IN R0.  UPDATE R5 TO POINT TO THE TOP OF THE VALUE
                           01C5     ;          STACK AGAIN.
                           01C5     ;
                           01C5     ; INPUTS:
                           01C5     ;
                           01C5     ;          R0      NUMBER OF BYTES TO OUTPUT (1 <= R0 <= 4)
                           01C5     ;          R5      POINTS TO TOP OF VALUE STACK
                           01C5     ;          R7      VALUE STACK POINTER
                           01C5     ;
                           01C5     ; OUTPUTS:
                           01C5     ;
                           01C5     ;          R5      POINTS TO NEW TOP OF STACK
                           01C5     ;          R7      UPDATED
                           01C5     ;          MAC$GL_PC INCREMENTED BY 1
                           01C5     ;
                           01C5     ;--
                           01C5
                           01C5     MAC$LIST_BYT_0::
  55   00000000'GF47  DE   01C5          MOVAL   G^MAC$AL_VALSTACK[R7],R5 ;POINT R5 TO TOP OF STACK
                           01CD     MAC$LIST_BYTES::
                 50   DD   01CD          PUSHL   R0                      ;COPY BYTE COUNT
                 50   DD   01CF          PUSHL   R0                      ;COPY ONCE MORE FOR GOOD LUCK
   53  00000000'GF    9E   01D1          MOVAB   G^MAC$GL_VALUE,R3       ;POINT TO RESULT AREA
             63  65   D0   01D8          MOVL    (R5),(R3)               ;COPY VALUE FROM TOP OF STACK
                 57   D7   01DB          DECL    R7                      ;'POP' THE STACK
                 75   D5   01DD          TSTL    -(R5)                   ;KEEP R5 POINTING TO TOP OF STACK
             50  83   9A   01DF     10$: MOVZBL  (R3)+,R0                ;GET NEXT BYTE
                 18   10   01E2          BSBB    MAC$LST_HEX_BYT         ;OUTPUT ONE BYTE
             F8  6E   F5   01E4          SOBGTR  (SP),10$                ;LOOP FOR ALL BYTES
                 8E   D5   01E7          TSTL    (SP)+                   ;CLEAN STACK
                           01E9          $INC_PC (SP)+                   ;INC. PC AND CLEAN STACK
       00000000'GF    D7   01EE          DECL    G^MAC$GL_LIST_PTR       ;BACK UP LISTING POINTER
   00000000'FF   20   90   01F4          MOVB    #^A/ /,@L^MAC$GL_LIST_PTR ;STORE A SPACE
                      05   01FB          RSB
```

```
                                      01FC   392 ;++
                                      01FC   393 ; FUNCTIONAL DESCRIPTION:
                                      01FC   394 ;
                                      01FC   395 ;       THIS ROUTINE OUTPUTS ONE BYTE TO THE LISTING FILE IN
                                      01FC   396 ;       HEX (TWO HEX DIGITS)
                                      01FC   397 ;
                                      01FC   398 ;--
                                      01FC   399
                                      01FC   400 MAC$LST_HEX_BYT::
                     50    50     9A  01FC   401         MOVZBL  R0,R0                   ;ENSURE REST OF WORD ZERO
                           50     DD  01FF   402         PUSHL   R0                      ;SAVE THE BYTE
      0001'8F     00000000'GF     B1  0201   403         CMPW    G^MAC$GL_LIST_PTR,#MAC$AB_LST_OP2+1 ; Time to dump buffer?
                           15     1A  020A   404         BGTRU   10$                     ; No If GTRU
                   0B 6B   27     E0  020C   405         BBS     #FLG$V_UPDFIL,(R11),5$  ; If updated file it is time now
      0001'8F     00000000'GF     B1  0210   406         CMPW    G^MAC$GL_LIST_PTR,#MAC$AB_LST_END+1 ;TIME TO DUMP THE BUFFER?
                           06     1A  0219   407         BGTRU   10$                     ;IF GTRU NO
                                      021B   408 5$:
                         004B     30  021B   409         BSBW    MAC$WRTLST
                     50    6E     D0  021E   410         MOVL    (SP),R0                 ;REFRESH THE BYTE
                           05     10  0221   411 10$:    BSBB    20$                     ;OUTPUT LOW 4 BITS
              50    8E   FC 8F    78  0223   412         ASHL    #-4,(SP)+,R0            ;POSITION TO GET HI 4 BITS
                  00000000'GF     D7  0228   413 20$:    DECL    G^MAC$GL_LIST_PTR       ;
                    50   F0 8F    8A  022E   414         BICB2   #^C<^XF>,R0             ;ISOLATE 4 BITS
  00000000'FF     0000023E'E0     90  0232   415         MOVB    L^HEX_TAB(R0),@L^MAC$GL_LIST_PTR ;CONVERT AND OUTPUT
                                  05  023D   416         RSB
                                      023E   417
42 41 39 38 37 36 35 34 33 32 31  30  023E   418 HEX_TAB:.ASCII  /0123456789ABCDEF/      ;CONVERSION FROM BINARY TO HEX
                      46 45 44    43  024A
```

```
                                   024E   420         .SBTTL  MAC$WRT_BLNKLIN WRITE A BLANK LINE TO LISTING
                                   024E   421
                                   024E   422 ;++
                                   024E   423 ; FUNCTIONAL DESCRIPTION:
                                   024E   424 ;
                                   024E   425 ;       THIS ROUTINE WRITES A BLANK LINE TO THE LISTING FILE
                                   024E   426 ;
                                   024E   427 ;--
                                   024E   428
                                   024E   429 MAC$WRT_3_BLNK::
                          03   DD  024E   430         PUSHL   #3                      ;SET TO WRITE 3 BLANK LINES
                          02   11  0250   431         BRB     WRT_BL
                                   0252   432
                                   0252   433 MAC$WRT_2_BLNK::
                          02   DD  0252   434         PUSHL   #2                      ;SET TO WRITE 2 BLANK LINES
                          06   10  0254   435 WRT_BL: BSBB    MAC$WRT_BLNKLIN         ;WRITE A BLANK LINE
                       FB 6E   F5  0256   436         SOBGTR  (SP),WRT_BL             ;LOOP FOR ALL
                          8E   D5  0259   437         TSTL    (SP)+                   ;CLEAN STACK
                               05  025B   438         RSB                             ;EXIT
                                   025C   439
                                   025C   440 MAC$WRT_BLNKLIN::
  00000000'GF    00000000'8F   D0  025C   441         MOVL    #-MAC$K_LIST_SIZE,G^MAC$GL_LINELN ;SET FOR NULL LINE
                          00   11  0267   442         BRB     MAC$WRTLST              ;WRITE LINE AND RETURN
                                   0269   443
                                   0269   444         .SBTTL  MAC$WRTLST      WRITE LINE TO LISTING FILE
                                   0269   445
                                   0269   446 ;++
                                   0269   447 ; FUNCTIONAL DESCRIPTION:
                                   0269   448 ;
                                   0269   449 ;       WRITES THE CURRENT LISTING LINE AND INITIALIZES FOR THE
                                   0269   450 ;       NEXT LINE.
                                   0269   451 ;
                                   0269   452 ;--
                                   0269   453
                                   0269   454 MAC$WRTLST::
           52    00000000'GF   D0  0269   455         MOVL    G^MAC$GL_LIST_IT,R2     ;GET CURRENT LISTING FLAG
                  24 6B   1B   E1  0270   456         BBC     #FLG$V_MACLTB,(R11),20$ ;BRANCH IF NOT EXPANDING A MACRO
              1D 00000005'EF   E8  0274   457         BLBS    L^LST$G_MACROXPAN+SYM$L_VAL,20$ ;EXPANDING--BRANCH IF LISTING
                                   027B   458                                                 ;MACRO EXPANSIONS
                          52   D4  027B   459         CLRL    R2                      ;CLEAR LISTING FLAG
              14 00000005'EF   E9  027D   460         BLBC    L^LST$G_MACROBIN+SYM$L_VAL,20$  ;BRANCH IF NOT LISTING
                                   0284   461                                                 ;MACRO BINARY
                  0D 6B   0C   E4  0284   462         BBSC    #FLG$V_MEBLST,(R11),10$ ;SPECIAL LIST FLAG ON?
  00000000'GF    00000000'GF   D1  0288   463         CMPL    G^MAC$GL_PC,G^MAC$GL_SAVE_PC ;SAME PC AS LAST TIME?
                          03   13  0293   464         BEQL    20$                     ;IF EQL YES
                     52   5E   D0  0295   465 10$:    MOVL    SP,R2                   ;NO--SET LISTING FLAG
           50    00000000'GF   D0  0298   466 20$:    MOVL    G^MAC$GL_LINELN,R0      ;GET LENGTH OF CURRENT LINE
            FFFFFFFF'E0   0C   91  029F   467         CMPB    #FF,L^MAC$AB_LINEBF-1(R0)  ;LAST CHARACTER A FORMFEED?
                          08   12  02A6   468         BNEQ    30$                     ;IF NEQ NO
                 00000000'GF   D7  02A8   469         DECL    G^MAC$GL_LINELN         ;YES--DON'T COUNT IT
                          0F   13  02AE   470         BEQL    40$                     ;IF EQL ZERO LENGTH LINE
                 00000000'GF   D5  02B0   471 30$:    TSTL    G^MAC$GL_LIST_LVL       ;CHECK THE LISTING LEVEL
                          07   19  02B6   472         BLSS    40$                     ;IF LSS DON'T LIST
                          09   14  02B8   473         BGTR    50$                     ;IF GTR LIST EVERYTHING
                     01   52   D1  02BA   474         CMPL    R2,#1                   ;DON'T LIST '.LIST/.NLIST' AT LEVEL 0
                          07   12  02BD   475         BNEQ    60$                     ;IF NEQ NOT .LIST/.NLIST
                          52   D4  02BF   476 40$:    CLRL    R2                      ;CLEAR LISTING FLAG
```

```
                03    11  02C1    477          BRB     60$
          52    5E    D0  02C3    478 50$:     MOVL    SP,R2                     ;SET LISTING FLAG
                52    D5  02C6    479 60$:     TSTL    R2                        ;LISTING FLAG CLEAR?
                56    13  02C8    480          BEQL    90$                       ;IF EQL YES--DO NO OUTPUT
       63 6B    09    E1  02CA    481          BBC     #FLG$V_LSTXST,(R11),MAC$WRT_LST_INI ;BRANCH IF NO LISTING FILE
                50    DD  02CE    482          PUSHL   R0                        ;SAVE INDEX INTO LINEBF
        00000000'GF   D5  02D0    483          TSTL    G^MAC$GL_LINE_CNT         ;AT THE END OF A PAGE
                04    15  02D6    484          BLEQ    70$                       ;IF LEQ YES--GO DO ONE
       03 6B    0A    E1  02D8    485          BBC     #FL^$V_NEWPND,(R11),80$ ;BRANCH IF NEW PAGE NOT PENDING
              00B8    30  02DC    486 70$:     BSBW    MAC$LST_PAG_HDR           ;YES--OUTPUT PAGE HEADER
  50    00000000'GF   D0  02DF    487 80$:     MOVL    G^MAC$GL_LINELN,R0        ;FIGURE SIZE OF LINE
     50     0000'8F   A0  02E6    488          ADDW2   #MAC$K_LIST_SIZE,R0       ;FIGURE TOTAL LENGTH OF LINE
  51    00000000'EF   9E  02EB    489          MOVAB   L^MAC$LIST_RAB,R1         ;POINT TO LISTING RAB
       22 A1    50    B0  02F2    490          MOVW    R0,RAB$W_RSZ(R1)          ;STORE RECORD SIZE
       23 6B    09    E1  02F6    491          BBC     #FLG$V_LSTXST,(R11),87$ ;BRANCH IF ERROR IN HEADER OUTPUT
28 A1   00000000'GF   9E  02FA    492          MOVAB   G^MAC$AB_LST_END,RAB$L_RBF(R1) ;STORE BUFFER ADDRESS
                          0302    493          $PUT    RAB=(R1),-                ;WRITE THE LINE TO THE LISTING
                          0302    494                  ERR=G^MAC$ERR_PUT
             03 50    E8  0311    495          BLBS    R0,85$                    ;BRANCH IF GOOD PUT
              FCE9'   30  0314    496          BSBW    MAC$CLOSE_LIST            ;CLOSE THE LISTING FILE ON ERROR
        00000000'GF   D7  0317    497 85$:     DECL    G^MAC$GL_LINE_CNT         ;DECREMENT LINES LEFT ON PAGE
                50  8ED0  031D    498 87$:     POPL    R0                        ;GET INDEX INTO LINEBF
  FFFFFFFF'E0   0C    91  0320    499 90$:     CMPB    #FF,L^MAC$AB_LINEBF-1(R0) ;FORM FEED?
                04    12  0327    500          BNEQ    100$                      ;IF NEQ NO
       00 6B    0A    E3  0329    501          BBCS    #FLG$V_NEWPND,(R11),100$ ;YES--FLAG NEW PAGE NEEDED
       00 6B    0C    E5  032D    502 100$:    BBCC    #FLG$V_MEBLST,(R11),110$ ;CLEAR SPECIAL LIST FLAG IF
                          0331    503                                            ;IT GOT SET AND WE WERE NOT
                          0331    504                                            ;IN A MACRO
                          0331    505 110$:
```

```
                                      0331  MAC$WRT_LST_INI::
                             38   BB  0331          PUSHR   #^M<R3,R4,R5>            ;SAVE REGISTERS
                    00000000'GF   D4  0333          CLRL    G^MAC$GL_LINELN          ;ZERO LENGTH OF LINE IN BUFFER
                   20    6B   00  2C  0339          MOVC5   #0,(R11),#^A/ /,-        ;FILL LISTING BUFFER WITH SPACES
                         0000'8F      033D                          #MAC$K_LIST_SIZE,-
                    00000000'GF       0340                          G^MAC$AB_LST_END
     00000000'GF    00000000'GF   9E  0345          MOVAB   G^MAC$AB_SEQ_NUM,G^MAC$GL_LIST_PTR ;INIT LISTING POINTER
                    00000000'GF   DD  0350          PUSHL   G^MAC$GL_VALUE           ;SAVE WHAT MIGHT BE HERE
                                      0356          $VPUSH  G^MAC$GL_PC              ;STACK THE PC
                         50   02  9A  0362          MOVZBL  #2,R0                    ;SET TO LIST TWO BYTES
                            FE5D  30  0365          BSBW    MAC$LIST_BYT_0           ;LIST THEM
                                      0368          $DEC_PC #2                       ;DON'T INCREMENT PC FOR IT THOUGH
                    00000000'GF 8ED0  036D          POPL    G^MAC$GL_VALUE           ;RESTORE VALUE
     00000000'GF    00000000'GF   D0  0374          MOVL    G^MAC$GL_PC,G^MAC$GL_SAVE_PC ;SAVE PC
            00000000'GF    FF 8F  98  037F          CVTBL   #-1,G^MAC$GL_LIST_IT     ;ASSUME LINE IS LISTED
                    00000000'GF   D7  0387          DECL    G^MAC$GL_LIST_PTR        ;BACK UP POINTER
              00000000'FF     20  90  038D          MOVB    #^A/ /,@L^MAC$GL_LIST_PTR ;STORE A SPACE THERE
                             38   BA  0394          POPR    #^M<R3,R4,R5>            ;RESTORE REGISTERS
                                  05  0396          RSB
```

```
                               0397               .SBTTL  MAC$LST_PAG_HDR WRITE NEW PAGE AND HEADER TO LISTING
                               0397  
                               0397  ;++
                               0397  ; FUNCTIONAL DESCRIPTION:
                               0397  ;
                               0397  ;            THIS ROUTINE OUTPUTS A PAGE MARK AND A NEW PAGE HEADER TO
                               0397  ;            THE LISTING FILE.
                               0397  ;
                               0397  ;--
                               0397  
                               0397  MAC$LST_PAG_HDR::
               00 6B    0A  E5 0397               BBCC      #FLG$V_NEWPND,(R11),10$ ;CLEAR NEW PAGE PENDING FLAG
  00000000'GF  00000000'GF  D0 039B  10$:         MOVL      G^MAC$GL_LN_PAGE,G^MAC$GL_LINE_CNT ; Reset lines/page
               03 6B    09  E0 03A6               BBS       #FLG$V_LSTXST,(R11),20$ ;BRANCH IF LISTING FILE
                      007F  31 03AA               BRW       50$                     ;NO LISTING FILE--GO AWAY
                        3F  BB 03AD  20$:         PUSHR     #^M<R0,R1,R2,R3,R4,R5>  ;SAVE REGISTERS
                               03AF  ;***;        $ASCTIM_S TIMBUF=MAC$AL_ATIM_DSC ;GET NEW TIME FOR PAGE HEADER
               00000000'GF  DD 03AF               PUSHL     G^MAC$GL_LIST_PTR       ;SAVE CURRENT LISTING POINTER
  00000000'GF  00000000'GF  9E 03B5               MOVAB     G^MAC$AB_HD_END,G^MAC$GL_LIST_PTR ;POINT TO WHERE PAGE # GOES
               00000000'GF  D6 03C0               INCL      G^MAC$GL_LPTPAG         ;BUMP PAGE COUNTER
           50  00000000'GF  D0 03C6               MOVL      G^MAC$GL_LPTPAG,R0      ;GET LISTING PAGE NUMBER
                      0077  30 03CD               BSBW      MAC$DEC_OUT_R2L         ;OUTPUT TO BUFFER
  00000000'GF  FFFFFFFF'GF  9E 03D0               MOVAB     G^MAC$AB_SBT_END-1,G^MAC$GL_LIST_PTR ;POINT TO WHERE SOURCE PAGE # G
         FFFFFFFF'GF    29  90 03DB               MOVB      #^A/)/,G^MAC$AB_SBT_END-1 ;STORE TERMINAL PAREN
           50  00000000'GF  D0 03E2               MOVL      G^MAC$GL_SRCPAG,R0      ;GET SOURCE PAGE NUMBER
                      005B  30 03E9               BSBW      MAC$DEC_OUT_R2L         ;OUTPUT SOURCE PAGE NUMBER
    50   00000000'GF    01  C3 03EC               SUBL3     #1,G^MAC$GL_LIST_PTR,R0 ;GET NEXT AVAIL SPOT
                  60    28  90 03F4               MOVB      #^A/(/,(R0)             ;STORE OPEN PAREN
           55  00000000'EF  9E 03F7               MOVAB     L^MAC$LIST_RAB,R5       ;POINT TO LISTING RAB
           22 A5   0001'8F  B0 03FE               MOVW      #MAC$K_HD_SIZE+1,RAB$W_RSZ(R5) ;STORE RECORD SIZE (INCLUDE FORMFEED)
        28 A5  00000000'GF  9E 0404               MOVAB     G^MAC$AB_HD_NEWPG,RAB$L_RBF(R5) ;AND RECORD ADDRESS
                        1F  10 040C               BSBB      60$                     ;WRITE FIRST LINE OF NEW PAGE
           22 A5   0000'8F  B0 040E               MOVW      #MAC$K_SBT_SIZ,RAB$W_RSZ(R5) ;LENGTH OF SUBTITLE LINE
        28 A5  00000000'GF  9E 0414               MOVAB     G^MAC$AB_SBT_IDNT,RAB$L_RBF(R5) ;AND ADDRESS
                        0F  10 041C               BSBB      60$                     ;WRITE SUBTITLE LINE
                     22 A5  B4 041E               CLRW      RAB$W_RSZ(R5)           ;WRITE BLANK LINE
                        0A  10 0421               BSBB      60$                     ;...
               00000000'GF8ED0 0423  30$:         POPL      G^MAC$GL_LIST_PTR       ;RESTORE LISTING POINTER
                        3F  BA 042A  40$:         POPR      #^M<R0,R1,R2,R3,R4,R5>  ;RESTORE REGISTERS
                            05 042C  50$:         RSB
                               042D  ;
                               042D  ; WRITE LINE TO LISTING -- R5 POINTS TO RAB
                               042D  ;
               15 6B    09  E1 042D  60$:         BBC       #FLG$V_LSTXST,(R11),70$ ;BRANCH IF LISTING DISABLED
                               0431               $PUT      RAB=(R5),ERR=G^MAC$ERR_PUT ;WRITE LINE TO LISTING
                     03 50  E8 0440               BLBS      R0,70$                  ;BRANCH IF GOOD PUT
                     FBBA'  30 0443               BSBW      MAC$CLOSE_LIST          ;ELSE CLOSE THE LISTING FILE
                            05 0446  70$:         RSB
```

```
                                   0447          .SBTTL  MAC$DEC_OUT_R2L OUTPUT DECIMAL NUMBER TO LISTING
                                   0447
                                   0447  ;++
                                   0447  ; FUNCTIONAL DESCRIPTION:
                                   0447  ;
                                   0447  ;       THIS ROUTINE OUTPUTS A 5 DIGIT OR LESS DECIMAL NUMBER
                                   0447  ;       TO THE LISTING FILE (IN REVERSE)
                                   0447  ;
                                   0447  ; INPUTS:
                                   0447  ;
                                   0447  ;       R0      NUMBER TO OUTPUT
                                   0447  ;
                                   0447  ;--
                                   0447
                                   0447  MAC$DEC_OUT_R2L::
                        38   BB    0447          PUSHR   #^M<R3,R4,R5>            ;SAVE REGISTERS
        0001869F 8F     50   D1    0449          CMPL    R0,#99999                ;CHECK FOR NUMBER TOO LARGE
                        07   1B    0450          BLEQU   10$                      ;IF LEQ NUMBER IS OK
   50   0001869F        8F   D0    0452          MOVL    #99999,R0                ;ELSE USE THE MAXIMUM
                  54    50   3C    0459  10$:    MOVZWL  R0,R4                    ;SET UP FOR THE EDIV
                        55   D4    045C          CLRL    R5                       ;CLEAR HIGH LONGWORD
  50    54        54    0A   7B    045E  20$:    EDIV    #10,R4,R4,R0             ;DO A DIVISION BY 10
                  50    30   80    0463          ADDB2   #^A/0/,R0                ;CONVERT DIGIT TO ASCII
                        4E   10    0466          BSBB    MAC$LST_CHAR             ;OUTPUT DIGIT TO LISTING BUFFER
                        54   D5    0468          TSTL    R4                       ;ARE WE DONE?
                        F2   12    046A          BNEQ    20$                      ;IF NEQ NO
                        38   BA    046C          POPR    #^M<R3,R4,R5>            ;YES--RESTORE REGISTERS
                             05    046E          RSB                              ;EXIT
```

```
                              046F          .SBTTL  MAC$DEC_OUT_L2X OUTPUT DECIMAL NUMBER LEFT TO RIGHT
                              046F
                              046F  ;++
                              046F  ; FUNCTIONAL DESCRIPTION:
                              046F  ;
                              046F  ;       THIS ROUTINE OUTPUTS A DECIMAL NUMBER LEFT-TO-RIGHT.
                              046F  ;
                              046F  ; INPUTS:
                              046F  ;
                              046F  ;       R0      NUMBER TO CONVERT
                              046F  ;       R1      OUTPUT POINTER
                              046F  ; OUTPUTS:
                              046F  ;
                              046F  ;       R0      UPDATED OUTPUT POINTER
                              046F  ;
                              046F  ;--
                              046F
                              046F  MAC$DEC_OUT_L2X::
             0078 8F     BB   046F          PUSHR   #^M<R3,R4,R5,R6>        ;SAVE REGISTERS
             56   51     D0   0473          MOVL    R1,R6                   ;SET OUTPUT POINTER
             54   50     D0   0476          MOVL    R0,R4                   ;AND NUMBER TO PRINT
                  55     D4   0479          CLRL    R5                      ;CLEAR HI WORD
                  26     10   047B          BSBB    DEC_OUT                 ;CONVERT AND OUTPUT THE NUMBER
             50   56     D0   047D          MOVL    R6,R0                   ;SET UPDATED POINTER
             0078 8F     BA   0480          POPR    #^M<R3,R4,R5,R6>        ;RESTORE REGISTERS
                         05   0484          RSB
                              0485
                              0485  ;++
                              0485  ; FUNCTIONAL DESCRIPTION:
                              0485  ;
                              0485  ;       THIS ROUTINE OUTPUTS A 5 DIGIT DECIMAL NUMBER IN R0 TO
                              0485  ;       THE LISTING FILE LEFT-TO-RIGHT.
                              0485  ;
                              0485  ;--
                              0485
                              0485  MAC$DEC_OUT_L2R::
             0078 8F     BB   0485          PUSHR   #^M<R3,R4,R5,R6>        ;SAVE REGISTERS
             54   50     D0   0489          MOVL    R0,R4                   ;COPY THE NUMBER
                  55     D4   048C          CLRL    R5                      ;CLEAR HIGH WORD
  56    00000000'GF      D0   048E          MOVL    G^MAC$GL_LIST_PTR,R6    ,GET THE LISTING POINTER
                  0C     10   0495          BSBB    DEC_OUT                 ;DO THE DIVISION
  00000000'GF     56     D0   0497          MOVL    R6,G^MAC$GL_LIST_PTR    ;UPDATE LISTING POINTER
             0078 8F     BA   049E          POPR    #^M<R3,R4,R5,R6>        ;RESTORE REGISTERS
                         05   04A2          RSB
                              04A3
                              04A3  DEC_OUT:
  50    54   54   0A     7B   04A3          EDIV    #10,R4,R4,R0            ;DIVIDE EM UP
        7E   50   30     81   04A8          ADDB3   #^A/0/,R0,-(SP)         ;CONVERT REMAINDER TO ASCII AND STACK IT
                  54     D5   04AC          TSTL    R4                      ;ARE WE DONE?
                  02     13   04AE          BEQL    20$                     ;IF EQL YES
                  F1     10   04B0          BSBB    DEC_OUT                 ;NO--RECURSE
             86   8E     90   04B2  20$:    MOVB    (SP)+,(R6)+             ;GET DIGIT BACK AND STORE IT
                         05   04B5          RSB                             ;RECURSE OR RETURN
                              04B6
                              04B6  ;++
                              04B6  ; FUNCTIONAL DESCRIPTION:
                              04B6  ;
                              04B6    662 ;       WRITE THE CHARACTER IN R0 INTO THE LISTING BUFFER FROM
                              04B6    663 ;       RIGHT TO LEFT.  THE POINTER INTO THE LISTING BUFFER IS
                              04B6    664 ;       DECREMENTED FIRST.
                              04B6    665 ;
                              04B6    666 ;--
                              04B6    667
                              04B6    668 MAC$LST_CHAR::
          00000000'GF     D7  04B6    669         DECL    G^MAC$GL_LIST_PTR       ;BACK UP THE POINTER
00000000'FF     50        90  04BC    670         MOVB    R0,@L^MAC$GL_LIST_PTR   ;PUT CHARACTER INTO LINE BUFFER
                          05  04C3    671         RSB
```

```
                                    04C4   673          .SBTTL  TERMINAL OUTPUT ROUTINES
                                    04C4   674
                                    04C4   675 ;++
                                    04C4   676 ; FUNCTIONAL DESCRIPTION:
                                    04C4   677 ;
                                    04C4   678 ;        THIS ROUTINE PRINTS THE LINE CONTAINED IN MAC$AB_LINEBF
                                    04C4   679 ;        AND MAC$AB_LST_END.  THE LENGTH OF THE LINE IS CALCULATED
                                    04C4   680 ;        AND THE LINE IS OUTPUT.
                                    04C4   681 ;
                                    04C4   682 ;--
                                    04C4   683
                                    04C4   684 MAC$TERM_BLANK::
                             50  D4  04C4   685          CLRL    R0                      ;ZERO FOR BLANK LINE
                         51  5E  D0  04C6   686          MOVL    SP,R1                   ;SHOULDN'T MATTER WHERE IT IS
                             11  11  04C9   687          BRB     MAC$PUT_TERM            ;GO OUTPUT AND RETURN
                                    04CB   688 MAC$WRITE_TERM::
  50  0000'8F   00000000'GF      A1  04CB   689          ADDW3   G^MAC$GL_LINELN,#MAC$K_LIST_SIZE,R0 ;COMPUTE SIZE OF LINE
            51   00000000'EF     9E  04D5   690          MOVAB   L^MAC$AB_LST_END,R1     ;GET ADDRESS OF BUFFER
                                    04DC   691
                                    04DC   692 MAC$PUT_TERM::
                             52  DD  04DC   693          PUSHL   R2                      ;SAVE R2
            52   00000000'EF     9E  04DE   694          MOVAB   L^MAC$TERM_RAB,R2       ;GET ADDRESS OF TERMINAL RAB
                    22 A2    50  B0  04E5   695          MOVW    R0,RAB$W_RSZ(R2)        ;SET THE RECORD SIZE IN THE RAB
                    28 A2    51  D0  04E9   696          MOVL    R1,RAB$L_RBF(R2)        ;SET THE RECORD ADDRESS
                                    04ED   697          $PUT    RAB=(R2),-              ;WRITE THE LINE TO THE TERMINAL
                                    04ED   698                  ERR=G^MAC$ERR_PUT
                          52 8ED0    04FC   699          POPL    R2                      ;RESTORE R2
                                 05  04FF   700          RSB
                                    0500   701
                                    0500   702          .END
```

```
$$.TMP1                 = 00000002                FLG$M_NTYPEPC           = 00000020
$$.TMP2                 = 00000062                FLG$M_NULCHR            = 00040000
$COUNT                  = 0000003B                FLG$M_OBJXST            = 00200000
ARG$K_SIZE              = 000003E8                FLG$M_OPNDCHK           = 00000100
AUD$K_SIZE              = 00000010                FLG$M_OPRND             = 00002000
BLNK                    = 00000020                FLG$M_OPTVFLIDX         = 00001000
CHR$M_COMMA_CR          = 00000020                FLG$M_ORDLST            = 00020000
CHR$M_ILL_CHR           = 00000040                FLG$M_P2                = 00004000
CHR$M_NUM_BER           = 00000010                FLG$M_RPTIRP            = 10000000
CHR$M_SPA_MSK           = 00000001                FLG$M_SEQFIL            = 02000000
CHR$M_SYM_CH1           = 00000008                FLG$M_SKAN              = 00008000
CHR$M_SYM_CHR           = 00000004                FLG$M_SPECOP            = 00000004
CHR$M_SYM_DLM           = 00000002                FLG$M_SPLALL            = 04000000
CHR$V_COMMA_CR          = 00000005                FLG$M_STOIMF            = 00040000
CHR$V_CVTLWC            = 00000061                FLG$M_SYM2COL           = 00000400
CHR$V_ILL_CHR           = 00000006                FLG$M_TOCFLG            = 00080000
CHR$V_NOCVT             = 0000007F                FLG$M_UPAFLG            = 00000010
CHR$V_NUM_BER           = 00000004                FLG$M_UPDFIL            = 00000080
CHR$V_SPA_MSK           = 00000000                FLG$M_UPMARG            = 00000040
CHR$V_SYM_CH1           = 00000003                FLG$M_XCRF              = 80000000
CHR$V_SYM_CHR           = 00000002                FLG$V_ALLCHR            = 00000000
CHR$V_SYM_DLM           = 00000001                FLG$V_BOL               = 00000001
CR                      = 0000000D                FLG$V_CHKLPND           = 00000014
DEC_OUT                   000004A3 R      04      FLG$V_COMPEXPR          = 00000002
ENB$G_OPTIONS             ********    X   04      FLG$V_CONT              = 00000003
FF                      = 0000000C                FLG$V_CRF               = 0000001E
FLG$M_ALLCHR            = 00000001                FLG$V_CRSEEN            = 00000020
FLG$M_BOL               = 00000002                FLG$V_DATRPT            = 00000004
FLG$M_CHKLPND           = 00100000                FLG$V_DBGOUT            = 0000002E
FLG$M_COMPEXPR          = 00000004                FLG$V_DLIMSTR           = 0000002F
FLG$M_CONT              = 00000008                FLG$V_ENDMCH            = 00000005
FLG$M_CRF               = 40000000                FLG$V_EVALEXPR          = 00000006
FLG$M_CRSEEN            = 00000001                FLG$V_EXPOPT            = 00000007
FLG$M_DATRPT            = 00000010                FLG$V_EXTERR            = 00000030
FLG$M_DBGOUT            = 00004000                FLG$V_EXTWRN            = 00000031
FLG$M_DLIMSTR           = 00008000                FLG$V_FIRSTLN           = 00000029
FLG$M_ENDMCH            = 00000020                FLG$V_IFSTAT            = 00000017
FLG$M_EVALEXPR          = 00000040                FLG$V_IIF               = 00000016
FLG$M_EXPOPT            = 00000080                FLG$V_INSERT            = 00000008
FLG$M_EXTERR            = 00010000                FLG$V_IRPC              = 0000001D
FLG$M_EXTWRN            = 00020000                FLG$V_LEXOP             = 00000021
FLG$M_FIRSTLN           = 00000200                FLG$V_LSTXST            = 00000009
FLG$M_IFSTAT            = 00800000                FLG$V_MAC2COL           = 0000002B
FLG$M_IIF               = 00400000                FLG$V_MACL              = 0000000B
FLG$M_INSERT            = 00000100                FLG$V_MACLTB            = 0000001B
FLG$M_IRPC              = 20000000                FLG$V_MACTXT            = 00000010
FLG$M_LEXOP             = 00000002                FLG$V_MEBLST            = 0000000C
FLG$M_LSTXST            = 00000200                FLG$V_MOREARG           = 0000002D
FLG$M_MAC2COL           = 00000800                FLG$V_MOREINP           = 00000023
FLG$M_MACL              = 00000800                FLG$V_NEWPND            = 0000000A
FLG$M_MACLTB            = 08000000                FLG$V_NOREF             = 00000018
FLG$M_MACTXT            = 00010000                FLG$V_NTYPEPC           = 00000025
FLG$M_MEBLST            = 00001000                FLG$V_NULCHR            = 00000032
FLG$M_MOREARG           = 00002000                FLG$V_OBJXST            = 00000015
FLG$M_MOREINP           = 00000008                FLG$V_OPNDCHK           = 00000028
FLG$M_NEWPND            = 00000400                FLG$V_OPRND             = 0000000D
FLG$M_NOREF             = 01000000                FLG$V_OPTVFLIDX         = 0000002C
```

```
FLG$V_ORDLST        = 00000011            MAC$GL_PC             ********   X   04
FLG$V_P2            = 0000000E            MAC$GL_PSECTPTR       ********   X   04
FLG$V_RPTIRP        = 0000001C            MAC$GL_RECTYP         ********   X   04
FLG$V_SEQFIL        = 00000019            MAC$GL_SAVE_PC        ********   X   04
FLG$V_SKAN          = 0000000F            MAC$GL_SAVE_SP        ********   X   04
FLG$V_SPECOP        = 00000022            MAC$GL_SRCPAG         ********   X   04
FLG$V_SPLALL        = 0000001A            MAC$GL_STOIMPTR       ********   X   04
FLG$V_STOIMF        = 00000012            MAC$GL_VALUE          ********   X   04
FLG$V_SYM2COL       = 0000002A            MAC$K_HD_SIZE         ********   X   04
FLG$V_TOCFLG        = 00000013            MAC$K_LIST_SIZE       ********   X   04
FLG$V_UPAFLG        = 00000024            MAC$K_SBT_SIZ         ********   X   04
FLG$V_UPDFIL        = 00000027            MAC$LIST_BYTES        000001CD RG    04
FLG$V_UPMARG        = 00000026            MAC$LIST_BYT_O        000001C5 RG    04
FLG$V_XCRF          = 0000001F            MAC$LIST_RAB          ********   X   04
HASHSZ              = 0000007F            MAC$LST_CHAR          000004B6 RG    04
HEX_TAB               0000023E R     04   MAC$LST_HEX_BYT       000001FC RG    04
HYPHEN              = 0000002D            MAC$LST_PAG_HDR       00000397 RG    04
INP$K_BUFSIZ        = 000003E8            MAC$OBJECT_RAB        ********   X   04
INT$K_BUFSIZ        = 000013F4            MAC$OBJHDROUT         ********   X   04
INT$K_BUFWRN        = 00001390            MAC$OUTOBJ            0000014B RG    04
LST$G_DIRLIST         ********   X   04   MAC$OUTOBJ_O          0000016C RG    04
LST$G_MACROBIN        ********   X   04   MAC$PASS2_DRIVR       00000000 RG    04
LST$G_MACROXPAN       ********   X   04   MAC$PASS_2_END        000000B0 RG    04
LST$K_BUFSIZ        = 00000086            MAC$PUT_TERM          000004DC RG    04
LST$K_L_P_PAGE      = 0000003C            MAC$STOIM             00000100 RG    04
LST$K_TITLE_SIZ     = 00000028            MAC$SYMNAMOUT         000000D6 RG    04
MAC$AB_HD_END         ********   X   04   MAC$TERM_BLANK        000004C4 RG    04
MAC$AB_HD_NEWPG       ********   X   04   MAC$TERM_RAB          ********   X   04
MAC$AB_LINEBF         ********   X   04   MAC$WRITE_TERM        000004CB RG    04
MAC$AB_LST_END        ********   X   04   MAC$WRTLST            00000269 RG    04
MAC$AB_LST_OP2        ********   X   04   MAC$WRTOBJ            00000174 RG    04
MAC$AB_OBJBUF         ********   X   04   MAC$WRT_2_BLNK        00000252 RG    04
MAC$AB_OBJWRN         ********   X   04   MAC$WRT_3_BLNK        0000024E RG    04
MAC$AB_SBT_END        ********   X   04   MAC$WRT_BLNKLIN       0000025C RG    04
MAC$AB_SBT_IDNT       ********   X   04   MAC$WRT_LST_INI       00000331 RG    04
MAC$AB_SEQ_NUM        ********   X   04   MAC_SUBSYS          = 0000007D
MAC$AL_VALSTACK       ********   X   04   OBJ$K_BUFSIZ        = 00000200
MAC$CHRBYT            00000140 RG    04   OPF$M_LASTOPR       = 00002000
MAC$CLOSE_LIST        ********   X   04   OPF$M_OPTEXP        = 00001000
MAC$CLS_DEL_OBJ       ********   X   04   OPF$V_LASTOPR       = 0000000D
MAC$DEC_OUT_L2R       00000485 RG    04   OPF$V_OPTEXP        = 0000000C
MAC$DEC_OUT_L2X       0000046F RG    04   P2$ADD                ********   X   03
MAC$DEC_OUT_R2L       00000447 RG    04   P2$AND                ********   X   03
MAC$EARLY_END         000000B8 RG    04   P2$ASH                ********   X   03
MAC$ERR_INTERN        ********   X   04   P2$ASN                ********   X   03
MAC$ERR_PUT           ********   X   04   P2$AUGPC              ********   X   03
MAC$FORM_LINENO       ********   X   04   P2$BDST               ********   X   03
MAC$GETLIN            ********   X   04   P2$CHKL               ********   X   03
MAC$GL_INTQUE         ********   X   04   P2$DISPATCH           00000000 RG    03
MAC$GL_LINELN         ********   X   04   P2$DIV                ********   X   03
MAC$GL_LINE_CNT       ********   X   04   P2$END                ********   X   03
MAC$GL_LIST_IT        ********   X   04   P2$EPT                ********   X   03
MAC$GL_LIST_LVL       ********   X   04   P2$ERR                ********   X   03
MAC$GL_LIST_PTR       ********   X   04   P2$ETX                ********   X   03
MAC$GL_LN_PAGE        ********   X   04   P2$FNEWL              ********   X   03
MAC$GL_LPTPAG         ********   X   04   P2$ILG                000000C2 RG    04
MAC$GL_OBJ_RCNT       ********   X   04   P2$INFO               ********   X   03
```

```
P2$K_MAXCOD                     = 0000003B                  PSC$M_BYTE                    = 00004000
P2$LGLAB                          ********   X    03        PSC$M_CON                     = FFFFFFFB
P2$MACL                           ********   X    03        PSC$M_DEFAULT                 = 000001C8
P2$MUL                            ********   X    03        PSC$M_EXE                     = 000000C0
P2$NEG                            ********   X    03        PSC$M_GBL                     = 00000010
P2$NEWL                           ********   X    03        PSC$M_LCL                     = FFFFFFEF
P2$NEWP                           ********   X    03        PSC$M_LIB                     = 00000002
P2$NOT                            ********   X    03        PSC$M_LONG                    = 00004800
P2$OP                             ********   X    03        PSC$M_NOEXE                   = FFFFFFBF
P2$OR                             ********   X    03        PSC$M_NOPIC                   = FFFFFFFE
P2$PRIL                           ********   X    03        PSC$M_NORD                    = FFFFFF7F
P2$PRT                            ********   X    03        PSC$M_NOSHR                   = FFFFFFDF
P2$PSECT                          ********   X    03        PSC$M_NOVEC                   = FFFFFDFF
P2$REDEF                          ********   X    03        PSC$M_NOWRT                   = FFFFFEFF
P2$REF                            ********   X    03        PSC$M_OVR                     = 00000004
P2$REST                           ********   X    03        PSC$M_PAGE                    = 00006400
P2$SAME                           ********   X    03        PSC$M_PIC                     = 00000001
P2$SAVE                           ********   X    03        PSC$M_QUAD                    = 00004C00
P2$SBTTL                          ********   X    03        PSC$M_RD                      = 00000080
P2$SETFLAG                        ********   X    03        PSC$M_REL                     = 00000008
P2$SETLONG                        ********   X    03        PSC$M_SHR                     = 00000020
P2$SPIC                           000000CC RG     04        PSC$M_USR                     = FFFFFFFD
P2$SPID                           ********   X    03        PSC$M_VEC                     = 00000200
P2$STIB                           ********   X    03        PSC$M_WORD                    = 00004400
P2$STIL                           ********   X    03        PSC$M_WRT                     = 00000180
P2$STIW                           ********   X    03        PSC$S_ALIGNMENT               = 00000004
P2$STKEPT                         ********   X    03        PSC$V_ALIGNFLG                = 0000000E
P2$STKG                           ********   X    03        PSC$V_ALIGNMENT               = 0000000A
P2$STKL                           ********   X    03        PSC$V_EXE                     = 00000006
P2$STKPC                          ********   X    03        PSC$V_GBL                     = 00000004
P2$STKS                           ********   X    03        PSC$V_LIB                     = 00000001
P2$STOB                           ********   X    03        PSC$V_OVR                     = 00000002
P2$STOL                           ********   X    03        PSC$V_PIC                     = 00000000
P2$STOW                           ********   X    03        PSC$V_RD                      = 00000007
P2$STRB                           ********   X    03        PSC$V_REL                     = 00000003
P2$STRL                           ********   X    03        PSC$V_SHR                     = 00000005
P2$STRSB                          ********   X    03        PSC$V_VEC                     = 00000009
P2$STRSW                          ********   X    03        PSC$V_WRT                     = 00000008
P2$STRW                           ********   X    03        PSC$W_FLAG                      00000009
P2$STSB                           ********   X    03        PSC$W_OPTIONS                   0000000D
P2$STSW                           ********   X    03        PSECT$MAIN                      ********   X    04
P2$SUB                            ********   X    03        RAB$L_RBF                     = 00000028
P2$SUME                           ********   X    03        RAB$W_RSZ                     = 00000022
P2$WRN                            ********   X    03        RDX$V_BINARY                  = 00000000
P2$XOR                            ********   X    03        RDX$V_DECIMAL                 = 00000002
PASS_2_LOOP                       00000065 R      04        RDX$V_DOUBLE                  = 00000005
PSC$B_NAME                        00000004                  RDX$V_FLOAT                   = 00000004
PSC$B_SEG                         0000000C                  RDX$V_GFLOAT                  = 00000006
PSC$B_UNUSED                      0000000B                  RDX$V_HEX                     = 00000003
PSC$K_BLKSIZ                      00000013                  RDX$V_HFLOAT                  = 00000007
PSC$K_NO_OPTNS                  = 0000000A                  RDX$V_OCTAL                   = 00000001
PSC$L_CURLOC                      0000000F                  REG$_PC                       = 0000000F
PSC$L_LINK                        00000000                  SEMI                          = 0000003B
PSC$L_MAXLGTH                     00000005                  STB$K_PG_MISS                 = 0000000A
PSC$M_ABS                       = FFFFFFF7                  SYM$B_NAME                      00000004
PSC$M_ALIGNFLG                  = 00004000                  SYM$B_SEG                       0000000C
PSC$M_ALLOPTNS                  = 000003FF                  SYM$B_TOKEN                     0000000B
```

```
SYM$K_BLKSIZ                    0000000D
SYM$K_MAXLEN                  = 0000001F
SYM$K_TWOCOL                  = 00000010
SYM$L_LINK                      00000000
SYM$L_VAL                       00000005
SYM$M_ABS                     = 00000010
SYM$M_ASN                     = 00000100
SYM$M_CRFO                    = 00002000
SYM$M_DEBUG                   = 00000020
SYM$M_DEF                     = 00000001
SYM$M_DELMAC                  = 00000200
SYM$M_EPT                     = 00000200
SYM$M_EXTRN                   = 00000008
SYM$M_GLOBL                   = 00000004
SYM$M_LOCAL                   = 00000040
SYM$M_ODBG                    = 00000400
SYM$M_REF                     = 00000080
SYM$M_RELPSECT                = 00000800
SYM$M_SUPR                    = 00004000
SYM$M_WEAK                    = 00000002
SYM$M_XCRF                    = 00001000
SYM$V_ABS                     = 00000004
SYM$V_ASN                     = 00000008
SYM$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
SYS$PUT                         ********  GX    04
TAB                           = 00000009
WRT_BL                          00000254 R      04
X1                            = 00000033
X2                            = 00080000

                                          +----------------+
                                          ! 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$RO_DATA                    000000EC  (   236.)  03 (  3.)  NOPIC    USR    CON    REL    GBL NOSHR NOEXE   RD   NOWRT NOVEC LONG
MAC$RO_CODE_P2                 00000500  (  1280.)  04 (  4.)  NOPIC    USR    CON    REL    GBL NOSHR   EXE   RD   NOWRT NOVEC LONG
```

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

Phase                    Page faults   CPU Time       Elapsed Time
-----                    -----------   --------       ------------
Initialization                  29     00:00:00.04    00:00:01.52
Command processing             104     00:00:00.33    00:00:02.42
Pass 1                         227     00:00:03.83    00:00:20.53
Symbol table sort                0     00:00:00.50    00:00:02.03
Pass 2                         146     00:00:01.27    00:00:05.01
Symbol table output             44     00:00:00.22    00:00:01.50
Psect synopsis output            2     00:00:00.02    00:00:00.02
Cross-reference output           0     00:00:00.00    00:00:00.00
Assembler run totals           554     00:00:06.21    00:00:33.03

The working set limit was 1350 pages.
36602 bytes (72 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 504 non-local and 48 local symbols.
702 source lines were read in Pass 1, producing 27 object records in Pass 2.
15 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                   8
_$255$DUA28:[SYSLIB]STARLET.MLB;2                    6
TOTALS (all libraries)                              14

568 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:P2DRVR/OBJ=OBJ$:P2DRVR MSRC$:P2DRVR/UPDATE=(ENH$:P2DRVR)+LIB$:MACRO/LIB
```

0227 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

MAIL

MAIL
MAP

P2DRVR
LIS

PARSER
LIS

RPTIRP
LIS

SCANER
LIS

TIMER
LIS

MAILCMDS
CLD

P2ACT2
LIS

SYMTAB
LIS