


```

FFFFFFFFF  IIIIII  NN    NN  IIIIII  SSSSSSSS  HH    HH
FFFFFFFFF  IIIIII  NN    NN  IIIIII  SSSSSSSS  HH    HH
FF         II     NN    NN  II       SS        HH    HH
FF         II     NN    NN  II       SS        HH    HH
FF         II     NN    NN  II       SS        HH    HH
FF         II     NNNN   NN  II       SS        HH    HH
FFFFFFFFF  II     NN    NN  II       SSSSSS    HHHHHHHHHH
FFFFFFFFF  II     NN    NN  II       SSSSSS    HHHHHHHHHH
FF         II     NN    NNNN  II       SS        HH    HH
FF         II     NN    NNNN  II       SS        HH    HH
FF         II     NN    NN    II       SS        HH    HH
FF         II     NN    NN    II       SS        HH    HH
FF         IIIIII  NN    NN  IIIIII  SSSSSSSS  HH    HH
FF         IIIIII  NN    NN  IIIIII! SSSSSSSS  HH    HH

```

```

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

```

(2)	101	DECLARATIONS
(3)	146	MAC\$SORT TABLE SORT SYMBOL TABLE ALPHABETICALLY
(4)	265	MAC\$FINISH ASM OUTPUT TRACEBACK, DEBUG INFO
(5)	396	PRINT SYMBOL TABLE IN ALPHABETICAL ORDER
(6)	498	PRINT PSECT SYNOPSIS
(7)	534	PRINT CROSS-REFERENCE IF REQUESTED
(8)	569	OUTPUT CPU AND ELAPSED TIME STATISTICS
(9)	611	PRINT MEMORY USE STATISTICS
(10)	660	PRINT MACRO LIBRARY USE STATISTICS
(11)	713	PRINT ERROR SUMMARY
(12)	786	PRINT COMMAND LINE
(13)	805	OUTPUT ONE PHASE OF CROSS REFERENCE
(14)	854	SET UP FOR NEW SUBTITLE AND CHECK NEW PAGE
(15)	896	OUTPUT SYMBOL NAME AND VALUE FOR DEBUG
(16)	943	OUTPUT PSECT RECORDS TO DEBUG
(17)	984	PRINT PSECT INFORMATION FOR 1 PSECT
(18)	1040	MATCH PSECT OPTIONS FOR PRINTING
(19)	1082	PRINT SYMBOL INFORMATION FOR ONE SYMBOL

```

0000 1      .TITLE MAC$FINISH      ROUTINES FOR FINISHING ASSEMBLY
0000 2      .IDENT 'V04-000'
0000 3
0000 4
0000 5 :*****
0000 6 :*
0000 7 :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :*  ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :*  TRANSFERRED.
0000 17 :*
0000 18 :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :*  CORPORATION.
0000 21 :*
0000 22 :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27
0000 28
0000 29 :++
0000 30 : FACILITY:      VAX MACRO ASSEMBLER OBJECT LIBRARY
0000 31
0000 32 : ABSTRACT:
0000 33
0000 34 : The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000 35 : modules for input to the VAX-11 LINKER.
0000 36
0000 37 : ENVIRONMENT:  USER MODE
0000 38
0000 39 : AUTHOR: Benn Schreiber, CREATION DATE: 25-AUG-78
0000 40
0000 41 : MODIFIED BY:
0000 42
0000 43 :      V03-004 MTR0035      Mike Rhodes      2-Aug-1983
0000 44 :      Correct psect definition in symbol table listing when the
0000 45 :      blank psect has been removed.
0000 46
0000 47 :      V03-003 MTR0033      Mike Rhodes      22-Apr-1983
0000 48 :      Allow the removal of the blank psect if it is not referenced.
0000 49
0000 50 :      V03-002 MTR0031      Mike Rhodes      19-Apr-1983
0000 51 :      Remove obsolete reference to $MAC_TIRCMDDEF macro.
0000 52
0000 53 :      V03-001 MTR0025      Mike Rhodes      8-Feb-1983
0000 54 :      Modify routines PRT_PSECT SYNOP, PRT CROSS REF, PRT RUN TIM,
0000 55 :      PRT_MLB_STATS, CREF_TREE_OUT, and NEW_SBT_CHK PAGE to write
0000 56 :      individual data records instead of having imbedded carriage
0000 57 :      control.

```

```

0000 58 :
0000 59 : V03.00 MTR0005 Mike Rhodes 15-Mar-1982
0000 60 : Modify routine MAC$DBG_PSECT to write a Position
0000 61 : Independent Data Reference (TIR$C_STO_PIDR) instead
0000 62 : of the Store Long Word (TIR$C_STO_LW).
0000 63 :
0000 64 : V02.16 PCG0003 Peter George 07-May-1981
0000 65 : Create bug which causes global symbols that have
0000 66 : been suppressed from the symbol table to have their
0000 67 : global symbol records suppressed.
0000 68 :
0000 69 : V02.15 PCG0001 Peter George 06-Feb-1981
0000 70 : Filter out creation of PSECT DST if ABS specified.
0000 71 :
0000 72 : V02.14 CNH0035 Chris Hume 12-Jun-1980
0000 73 : Prevented extraneous symbols at the end of two column (31
0000 74 : character) symbol table output.
0000 75 :
0000 76 : V01.13 RN0022 R. Newland 31-Oct-1979
0000 77 : Translate SY$SLP_LINES to set lines/page
0000 78 :
0000 79 : V01.12 RN0013 R. Newland 27-Sep-1979
0000 80 : Use new symbols for PSECT options processing
0000 81 :
0000 82 : V01.11 RN0011 R. Newland 11-Sep-1979
0000 83 : New Librarian support
0000 84 :
0000 85 : V01.10 RN0008 R. Newland 29-Aug-1979
0000 86 : 31 character symbols
0000 87 :
0000 88 : V01.09 RN0005 R. Newland 13-Aug-1979
0000 89 : Variable symbol name storage
0000 90 :
0000 91 : V01.08 RN0002 R. Newland 01-Feb-1979
0000 92 : Changes for Source Update Merge, information messages
0000 93 : count and line number format
0000 94 :
0000 95 :
0000 96 : V01.06 BLS21346 B. Schreiber 29-DEC-1978
0000 97 : Remove restriction that symbol must be referenced
0000 98 : and not absolute to be output to debugger.
0000 99 :--

```

```

0000 101      .SBTTL  DECLARATIONS
0000 102      :
0000 103      : INCLUDE FILES:
0000 104      :
0000 105      :
0000 106      :
0000 107      : MACROS:
0000 108      :
0000 109      :
0000 110      $MAC_SYMBLKDEF      ;SYMBOL BLOCK DEFINITIONS
0000 111      $MAC_MLFDEF        ; Define MLF offsets
0177 112      $MAC_GENVALDEF     ;GENERAL VALUES
0177 113      $MAC_CTLFLGDEF    ;CONTROL FLAGS
0177 114      $MAC_CRFLAGDEF    ;DEFINE CRF CONTROL FLAGS
0177 115      $MAC_OBJCODDEF    ;DEFINE OBJECT CODE COMMANDS
0000 116      $FABDEF           ;DEFINE FAB OFFSETS
0000 117      $CRFDEF           ; Define CRF offsets
0000 118      :
0000 119      :
0000 120      : LOCAL SYMBOLS
0000 121      :
0000 122      :
000000B8 0000 123 DBG$C_PSECT      =      184      ;PSECT NAME (1+11+N-ENTRY LENGTH)
000000BA 0000 124 DBG$C_SYMBOL    =      186      ;SYMBOL NAME (1+7+N)
000000BC 0000 125 DBG$C_MODULE   =      188      ;MODULE NAME (1+7+N)
000000BD 0000 126 DBG$C_MEND     =      189      ;MODULE END (1+1)
000000BE 0000 127 DBG$C_ROUTINE  =      190      ;ROUTINE NAME (1+7+N)
00000000 0000 128 DBG$C_LIT_DAT   =           0      ;LITERAL DATA
00000001 0000 129 DBG$C_REL_DAT   =           1      ;RELOCATABLE DATA
0000000C 0000 130 DBG$K_PSECT_LEN  =          12      ;CONSTANT LENGTH OF PSECT ENTRY
00000008 0000 131 DBG$K_SYMBOL_LN =           8      ;CONSTANT LENGTH OF SYMBOL ENTRY
00000008 0000 132 DBG$K_MODULE_LN=           8      ;CONSTANT LENGTH OF MODULE ENTRY
00000002 0000 133 DBG$K_MEND_LEN  =           2      ;CONSTANT LENGTH OF MODULE END ENTRY
00000008 0000 134 DBG$K_ROUTIN_LN =           8      ;CONSTANT LENGTH OF ROUTINE ENTRY
0000 135      :
0000 136      :
0000 137      : LOCAL DATA
0000 138      :
0000 139      :
00000000 0000 140      .PSECT  MAC$RW_DATA,NOEXE, LONG
0000 141      :
00000000' 0000 142 MAC$G_ERRBFDES::          ;TO FAO THE ERROR LINE #'S
00000000 0004 143      .LONG  MAC$AB_LINE_END-MAC$AB_LST_END ;SIZE OF BUFFER
00000000 0004 144      .LONG  0              ;FILLED IN WITH ADDRESS

```

```

0008 146      .SBTTL MAC$SORT_TABLE SORT SYMBOL TABLE ALPHABETICALLY
0008 147
0008 148 :++
0008 149 : FUNCTIONAL DESCRIPTION:
0008 150 :
0008 151 :     THIS ROUTINE WILL SORT THE SYMBOL TABLE INTO AN ALPHABETIC
0008 152 :     LINKED LIST.
0008 153 :
0008 154 : INPUTS:
0008 155 :
0008 156 :     R9     SYMBOL HASH TABLE ADDRESS
0008 157 :     R8     SYMBOL HASH TABLE SIZE
0008 158 :
0008 159 : OUTPUTS:
0008 160 :
0008 161 :     MAC$GL_LINK_PTR POINTER TO LINKED LIST OF SYMBOLS
0008 162 :
0008 163 :--
0008 164
00000000 165      .PSECT MAC$RO_CODE_P3,NOWRT,GBL, LONG
0000 166
0000 167 MAC$SORT TABLE::
0000 168      PUSHAB W*MAC$GO RNT SRT      ;STACK TIME BLOCK ADDRESS
0000 169      CALLS #1,W*MAC$TIMER_ON    ;BEGIN TIMING SYMBOL TABLE SORT
0009 170 :
0009 171 : THE FIRST STEP IS TO REMOVE ALL EMPTY SUBLIST LISTHEADS FROM THE TABLE
0009 172 :
0009 173      MOVL R9,R7      ;COPY TABLE ADDRESS
0009 174      MOVL R7,R6
0009 175 10$: TSTL (R7)+      ;LIST HEAD EMPTY?
0009 176      BEQL 20$      ;IF EQL YES
0009 177      MOVL -4(R7),(R6)+ ;NO--MOVE THE LISTHEAD ENTRY
0009 178 20$: SOBGTR R8,10$ ;SCAN ALL LISTHEADS
0009 179      CLRL W*MAC$GL_LINK_PTR ;ASSUME NO ENTRIES
0009 180      SUBL3 R9,R6,R5 ;CALCULATE SIZE OF TABLE IN BYTES
0009 181      BNEQ SORT_LISTS ;BRANCH IF THERE ARE SOME ENTRIES
0009 182      BRW SORT_EXIT ;ELSE GO EXIT
0009 183 :
0009 184 : NEXT, WE SORT THE SUBLISTS SUCH THAT THE FIRST SYMBOL IN EACH SUBLIST
0009 185 : IS LESS THAN THE FIRST SYMBOL OF THE SUCCEEDING SUBLIST.
0009 186 :
0009 187 SORT_LISTS:
0009 188      MOVL R9,R8      ;COPY TABLE ADDRESS
0009 189 10$: MOVL (R8)+,R7 ;GET ADDRESS OF NEXT SYMBOL
0009 190      CMPL R8,R6      ;ANY MORE LISTHEADS IN TABLE?
0009 191      BEQL MERGE_LISTS ;IF EQL NO
0009 192      MOVL (R8),R5 ;GET ADDRESS OF NEXT
0009 193      MOVZBL SYMSB_NAME(R7),R0 ; Get offset to name
0009 194      SUBL3 R0,R7,R0 ; and form address of count/name
0009 195      MOVZBL (R0)+,R2 ; Get count and advance pointer to name
0009 196      MOVZBL SYMSB_NAME(R5),R1 ; Get offset to name
0009 197      SUBL3 R1,R5,R1 ; and form address of count/name
0009 198      MOVZBL (R1)+,R3 ; Get count and advance pointer to name
0009 199      CMPC5 R2,(R0),#0,R3,(R1) ; Compare symbols
0009 200      BLSSU 10$ ;IF LSS THEN ORDER CORRECT
0009 201      MOVL R7,(R8) ;INVERT LISTHEAD ENTRIES
0009 202      MOVL R5,-(R8) ;...

```

0000'CF 9F
0000'CF 01 FB

57 59 D0
56 57 D0
87 D5
04 13
86 FC A7 D0
F5 58 F5
55 0000'CF D4
56 59 C3
03 12
00CE 31

58 54 D0
57 86 D0
56 58 D1
30 13
55 68 D0
50 04 A7 9A
50 57 50 C3
52 80 9A
51 04 A5 9A
51 55 51 C3
53 81 9A
61 53 00 60 52 2D
D7 1F
68 57 D0
78 55 D0

61 53 00

```

59 58 D1 0059 203      CMPL  R8,R9      ;AT FRONT OF LISTHEAD TABLE?
      CC 13 005C 204      BEQL  10$      ;IF EQL YES
      78 D5 005E 205      TSTL  -(R8)     ;NO--BACKUP ONE ENTRY
      C8 11 0060 206      BRB   10$
      0062 207      ;
      0062 208      ; NOW MERGE THE SUBLISTS INTO ONE MASTER SORTED LIST
      0062 209      ;
      0062 210 MERGE_LISTS:
58 0000'CF 9E 0062 211      MOVAB  W*MAC$GL_LINK_PTR,R8 ;POINT TO MERGE LISTHEAD
      57 59 D0 0067 212      MOVL  R9,R7      ;COPY LISTHEAD TABLE POINTER
      55 87 D0 006A 213 10$: MOVL  (R7)+,R5 ;GET ADDRESS OF NEXT SYMBOL
      03 12 006D 214      BNEQ  12$      ; If NEQ more in this sublist
      007C 31 006F 215      BRW   70$
      0072 216 12$:
2C 09 A5 06 E0 0072 217      BBS   #SYMSV_LOCAL,SYMSW_FLAG(R5),20$ ;BR IF LOCAL SYMBOL
0A 09 A5 00 E0 0077 218      BBS   #SYMSV_DEF,SYMSW_FLAG(R5),15$ ;BRANCH IF DEFINED
05 09 A5 03 E0 007C 219      BBS   #SYMSV_EXTRN,SYMSW_FLAG(R5),15$ ;BRANCH IF DECLARED EXTERNAL
21 09 A5 07 E1 0081 220      BBC   #SYMSV_REF,SYMSW_FLAG(R5),30$ ;BRANCH IF NOT REFERENCED
      0086 221      ;(SYMBOL WAS USED IN .NTYPE DIRECTIVE
      0086 222      ;AND WAS NOT REF. OTHERWISE)
      0000'CF D6 0086 223 15$: INCL  W*MAC$GL_SYM_NLOC ;COUNT NON-LOCAL SYMBOL
      68 55 D0 008A 224      MOVL  R5,SYMSL_LINK(R8) ; Link new symbol to old last
      56 55 D0 008D 225      MOVL  R5,R8      ;MAKE NEW LAST SYMBOL
      0090 226      ;
      0090 227      ; IF .ENABLE GLOBAL AND SYMBOL IS UNDEFINED, DECLARE IT EXTERNAL NOW.
      0090 228      ;
10 00000005'EF E9 0090 229      BLBC  L^ENBSG GLOBAL+SYMSL_VAL,30$ ;BRANCH IF DISABLE GLOBAL
0B 09 A5 00 E0 0097 230      BBS   #SYMSV_DEF,SYMSW_FLAG(R5),30$ ;BRANCH IF SYMBOL DEFINED
06 09 A5 03 E3 009C 231      BBCS  #SYMSV_EXTRN,SYMSW_FLAG(R5),30$ ;DECLARE SYMBOL EXTERNAL
      04 11 00A1 232      BRB   30$      ;GO AHEAD
      0000'CF D6 00A3 233 20$: INCL  W*MAC$GL_SYM_LOCL ;COUNT A LOCAL SYMBOL
      77 65 D0 00A7 234 30$: MOVL  SYMSL_LINK(R5),-(R7) ; Remove symbol from list
      02 12 00AA 235      BNEQ  40$      ;IF NEQ THEN MORE IN LIST
      87 D5 00AC 236      TSTL  (R7)+ ;ADVANCE LISTHEAD POINTER
      56 57 D1 00AE 237 40$: CMPL  R7,R6 ;ANY MORE SUBLISTS TO CONSIDER?
      40 13 00B1 238      BEQL  80$      ;IF EQL NO
      7E 57 7D 00B3 239      MOVQ  R7,-(SP) ;SAVE MERGE PARMS (R7/R8)
      58 87 D0 00B6 240 50$: MOVL  (R7)+,R8 ;GET ADDRESS OF FIRST SUBLIST ENTRY
      56 57 D1 00B9 241      CMPL  R7,R6 ;ANY MORE SUBLISTS TO CONSIDER?
      2A 13 00BC 242      BEQL  60$      ;IF EQL NO
      55 67 D0 00BE 243      MOVL  (R7),R5 ;YES--GET ADDRESS OF FIRST SUBLIST ENTRY
      50 04 A8 9A 00C1 244      MOVZBL SYMSB_NAME(R8),R0 ; Get offset to first name
      50 58 50 C3 00C5 245      SUBL3  R0,R8,R0 ; and form address of count/name
      52 80 9A 00C9 246      MOVZBL (R0)+,R2 ; Get count and advance pointer to name
      51 04 A5 9A 00CC 247      MOVZBL SYMSB_NAME(R5),R1 ; Get offset to second name
      51 55 51 C3 00D0 248      SUBL3  R1,R5,R1 ; and form address of count/name
      53 81 9A 00D4 249      MOVZBL (R1)+,R3 ; Get count and advance pointer to name
61 53 00 60 52 2D 00D7 250      CMPQ  R2,(R0),#0,R3,(R1) ; Compare symbols
      09 1F 00DD 251      BLSSU  60$      ;IF LSS ORDER CORRECT
      67 58 D0 00DF 252      MOVL  R8,(R7) ;SWAP SUBLISTS
      FC A7 55 D0 00E2 253      MOVL  R5,-4(R7)
      CE 11 00E6 254      BRB   50$
      57 8E 7D 00E8 255 60$: MOVQ  (SP)+,R7 ;RESTORE MERGE PARAMETERS (R7/R8)
      FF 7C 31 00EB 256 65$: BRW   10$      ; Continue merge
      56 57 D1 00EE 257 70$: CMPL  R7,R6 ;ANY MORE SUBLISTS TO CONSIDER?
      F8 12 00F1 258      BNEQ  65$      ; If NEQ yes
      68 D4 00F3 259 80$: CLRL  SYMSL_LINK(R8) ; Clear forward link of last entry

```

```
0000'CF 01 00F5 260 SORT_EXIT:  
          9F 00F5 261 PUSHAB W^MAC$GQ RNT_SRT ;STACK TIMING BLOCK ADDRESS  
          FB 00F9 262 CALLS #1,W^MAC$TIMER_OFF ;STOP TIMING SYMBOL TABLE SORT  
          05 00FE 263 RSB
```

```

O0FF 265      .SBTTL MAC$FINISH_ASM OUTPUT TRACEBACK, DEBUG INFO
O0FF 266
O0FF 267      :++
O0FF 268      : FUNCTIONAL DESCRIPTION:
O0FF 269
O0FF 270      : THIS ROUTINE IS CALLED AT THE END OF THE ASSEMBLY TO FINISH
O0FF 271      : THINGS UP. IF ENABLED, TRACEBACK AND DEBUG INFORMATION IS
O0FF 272      : OUTPUT TO THE OBJECT FILE. THE SYMBOL TABLE LISTING IS PRINTED.
O0FF 273      :
O0FF 274      :--
O0FF 275
O0FF 276 MAC$FINISH_ASM::
0380 8F BB O0FF 277      PUSHR      #*M<R7,R8,R9>      ;SAVE REGISTERS
      57 D4 0103 278      CLRL       R7              ;RESET STACK POINTER
03 6B 15 E0 0105 279      BBS       #FLG$V_OBJXST,(R11),5$    ; Branch if there is an object file
      0084 31 0109 280      BRW       85$              ; No object file
      010C 281 5$:
7D 00000005'EF E9 010C 282      BLBC     L*ENBSG_TRACEBACK+SYMSL_VAL,85$ ; Branch if disable traceback
0000'CF 05 9A 0113 283      MOVZBL  #OBJ$C_TBT,W*MAC$GL_RECTYP ;TRACEBACK RECORD TYPE
      FEES' 30 0118 284      BSBW    MAC$WRTOBJ      ;WRITE OUT LAST TIR RECORD
55 0000'CF 9E 011B 285      MOVAB   W*MAC$AB_TITLE,R5      ;POINT TO TITLE STRING
      65 95 0120 286      TSTB   (R5)                ;WAS TITLE SUPPLIED?
      07 12 0122 287      BNEQ   10$              ;IF NEQ YES--GO USE IT
55 00000000'EF 9E 0124 288      MOVAB   L*MAC$AB_DEF_TITL,R5    ;POINT TO DEFAULT TITLE STRING
50 07 65 81 012B 289 10$: ADDB3   (R5),#DBG$K_MODULE_LN-1,R0 ;FIGURE LENGTH OF ENTRY
      FECE' 30 012F 290      BSBW    MAC$STOIM        ;STORE INTO OBJECT CODE
50 BC 8F 9A 0132 291      MOVZBL  #DBG$C_MODULE,R0      ;MODULE NAME
      FEC7' 30 0136 292      BSBW    MAC$STOIM        ;STORE IT
      54 05 9A 0139 293      MOVZBL  #5,R4              ;SET A LOOP COUNT
      50 D4 013C 294 20$: CLRL     R0              ;STRUCTURE LEVEL AND 'MACRO?(?)
      FEBF' 30 013E 295      BSBW    MAC$STOIM
      F8 54 F5 0141 296      SOBGTR  R4,20$            ;DO 5
54 85 9A 0144 297      MOVZBL  (R5)+,R4          ;GET CHARACTER COUNT
50 54 D0 0147 298      MOVL    R4,R0            ;COPY TO STORE IT
      FEB3' 30 014A 299      BSBW    MAC$STOIM
50 85 9A 014D 300 40$: MOVZBL  (R5)+,R0          ;GET CHAR OF TITLE
      FEAD' 30 0150 301      BSBW    MAC$STOIM        ;STORE IMMEDIATE
      F7 54 F5 0153 302      SOBGTR  R4,40$
      0156 303      :
      0156 304      : SEARCH SYMBOL TABLE AND EMIT ENTRY POINTS TO THE DEBUGGER
      0156 305      :
56 0000'CF D0 0156 306      MOVL    W*MAC$GL_LINK_PTR+SYMSL_LINK,R6 ; Point to linked symbol list
      2B 13 015B 307      BEQL    80$              ;IF EQL THERE IS NO LIST
21 09 A6 09 E1 015D 308 60$: BBC     #SYMSV_EPT,SYMSW_FLAG(R6),70$ ;BRANCH IF NOT EPT
50 50 04 A6 9A 0162 309      MOVZBL  SYMSB_NAME(R6),R0    ; Get offset to name
50 56 50 C3 0166 310      SUBL3   R0,R6,R0          ; Form address of count/name
      50 07 81 016A 311      ADDB3   #DBG$K_SYMBOL_LN-1,- ; Figure length of entry
      50 60 30 016C 312      (R0),R0 ; putting result in R0
      FE8F' 30 016E 313      BSBW    MAC$STOIM
50 BE 8F 9A 0171 314      MOVZBL  #DBG$C_ROUTINE,R0    ;TYPE
      FE88' 30 0175 315      BSBW    MAC$STOIM
      50 D4 0178 316      CLRL    R0              ;ZERO BYTE
      FE83' 30 017A 317      BSBW    MAC$STOIM
      06E6 30 017D 318      BSBW    MAC$DBG_VAL_OUT    ;OUTPUT ADDRESS
      06D1 30 0180 319      BSBW    MAC$DBG_NAM_OUT    ;OUTPUT NAME
56 66 D0 0183 320 70$: MOVL    SYMSL_LINK(R6),R6 ; Link to next symbol
      D5 12 0186 321      BNEQ   60$              ;IF THERE IS ONE, GO PROCESS IT

```

```

0188 322 :
0188 323 : OUPUT LABLES AND REFERENCED SYMBOLS FOR DEBUGGER
0188 324 :
0000'CF 04 9A 0188 325 80$: MOVZBL #OBJ$C_DBG,W*MAC$GL_RECTYP ;SET DEBUG RECORD TYPE
FE70' 30 018D 326 BSBW MAC$WRTOBJ ;WRITE OUT THE TRACEBACK RECORDS
56 0000'CF D0 0190 327 85$: MOVL W*MAC$GL_LINK_PTR,R6 ;POINT TO SYMBOL LIST AGAIN
64 13 0195 328 BEQL 150$ ;IF EQL THERE IS NO LIST
0197 329 :
0197 330 : BACK HERE FOR EACH SYMBOL IN THE LIST
0197 331 :
53 09 A6 3C 0197 332 90$: MOVZWL SYMSW FLAG(R6),R3 ;GET FLAGS FOR SYMBOL
57 53 05 E1 019B 333 BBC #SYMSV_DEBUG,R3,140$ ;BRANCH IF NO DEBUG ATTRIBUTE
53 53 00 E1 019F 334 BBC #SYMSV_DEF,R3,140$ ;BRANCH IF NOT DEFINED
01A3 335 :
01A3 336 : SYMBOL HAS DEBUG ATTRIBUTE AND IS DEFINED.
01A3 337 :
09 A6 0400 8F A8 01A3 338 100$: BISW2 #SYMSM_ODBG,SYMSW FLAG(R6) ;SET DEBUG OUTPUT FLAG FOR SYMBOL
49 53 09 E0 01A9 339 BBS #SYMSV_EPT,R3,140$ ;BRANCH IF THIS IS AN ENTRY POINT
45 68 15 E1 01AD 340 BBC #FLGSV_OBJXST,(R11),140$ ;BRANCH IF THERE IS NO OBJECT FILE
3E 00000005'EF E9 01B1 341 BLBC L*ENBSG_TRACEBACK+SYMSL_VAL,140$ ;BRANCH IF DISABLE TRACEBACK
50 04 A6 9A 01B8 342 MOVZBL SYMSB_NAME(R6),R0 ; Get offset to name
50 56 50 C3 01BC 343 SUBL3 R0,R6,R0 ; Form address of count/name
81 01C0 344 ADDB3 #DBG$K_SYMBOL_LN-1,- ; Figure length of entry
50 60 01C2 345 (R0),R0 ; putting result in R0
FE39' 30 01C4 346 BSBW MAC$STOIM
50 BA 8F 9A 01C7 347 MOVZBL #DBG$C_SYMBOL,R0 ;ASSUME A LABEL
FE32' 30 01CB 348 BSBW MAC$STOIM ;STORE INTO OBJECT CODE
OB 52 04 E0 01CE 349 BBS #SYMSV_ABS,R2,110$ ;BRANCH IF SYMBOL IS ABSOLUTE
01D2 350 :
01D2 351 : SYMBOL IS RELOCATABLE
01D2 352 :
50 01 9A 01D2 353 MOVZBL #DBG$C_REL_DAT,R0 ;RELOCATABLE DATA
FE28' 30 01D5 354 BSBW MAC$STOIM
068B 30 01D8 355 BSBW MAC$DBG_VAL_OUT ;OUTPUT SYMBOL ADDRESS
16 11 01DB 356 BRB 130$
01DD 357 :
01DD 358 : SYMBOL IS ABSOLUTE
01DD 359 :
50 00 9A 01DD 360 110$: MOVZBL #DBG$C_LIT_DAT,R0 ;LITERAL DATA
FE1D' 30 01E0 361 BSBW MAC$STOIM
55 05 A6 9E 01E3 362 MOVAB SYMSL_VAL(R6),R5 ;POINT TO SYMBOL VALUE
54 04 9A 01E7 363 MOVZBL #4,R4 ;LOOP COUNT
50 85 90 01EA 364 120$: MOVB (R5)+,R0 ;GET BYTE OF VALUE
FE10' 30 01ED 365 BSBW MAC$STOIM ;STORE INTO OBJECT CODE
F7 54 F5 01F0 366 SOBGTR R4,120$
01F3 367 :
01F3 368 : OUTPUT SYMBOL NAME
01F3 369 :
065E 30 01F3 370 130$: BSBW MAC$DBG_NAM_OUT ;SEND NAME TO OBJECT FILE
56 66 D0 01F6 371 140$: MOVL SYMSL_LINK(R6),R6 ; Link to next symbol
9C 12 01F9 372 BNEQ 90$ ;BRANCH IF THERE IS ONE
40 6B 15 E1 01FB 373 150$: BBC #FLGSV_OBJXST,(R11),PRINT SYM TABLE ;SKIP IF NO OBJECT FILE
39 000000C5'EF E9 01FF 374 BLBC L*ENBSG_TRACEBACK+SYMSL_VAL,PRINT SYM TABLE ;SKIP IF DISABLE TRACEBA
0000'CF 05 9A 0206 375 MOVZBL #OBJ$C_TBT,W*MAC$GL_RECTYP ;SET FOR TRACEBACK RECORD TYPE
FDF2' 30 020B 376 BSBW MAC$WRTOBJ ;WRITE OUT DEBUG RECORD
020E 377 :
020E 378 : OUTPUT PSECT NAMES AND END OF MODULE

```

```
56 0000'CF 9E 020E 379 ;  
03 09 A6 07 E1 0213 380 MOVAB W^PSECT$BLANK,R6 ;BLANK PSECT  
0677 30 0218 381 BBC #SYMSV REF,SYMSW_FLAG(R6),155$ ;HAS IT BEEN REF'D?  
56 0000'CF D0 021B 382 BSBW MAC$DBG_PSECT ;YES, OUTPUT INFORMATION  
0D 13 0220 383 155$: MOVL W^MAC$GC_PSC_LIST,R6 ;GET FIRST PSECT ADDRESS  
03 0D A6 03 E1 0222 384 BEQL 170$ ;IF EQL NONE  
0227 385 160$: BBC #PSC$V REL, - ;SUPPRESS ABS PSECTS  
0227 386 PSC$W_OPTIONS(R6),165$  
56 0668 30 0227 387 BSBW MAC$DBG_PSECT ;OUPUT INFO  
56 66 D0 022A 388 165$: MOVL SYMSL_LINK(R6),R6 ; Link to next symbol  
F3 12 022D 389 BNEQ 160$ ;IF NEQ THERE IS MORE  
50 01 9A 022F 390 170$: MOVZBL #1,R0 ;STORE LENGTH  
FDCB' 30 0232 391 BSBW MAC$STOIM  
50 BD 8F 9A 0235 392 MOVZBL #DBG$C_MEND,R0 ;MODULE END  
FDC4' 30 0239 393 BSBW MAC$STOIM  
FDC1' 30 023C 394 BSBW MAC$WRTOBJ ;WRITE OUT FINAL TRACEBACK RECORD
```

```

                                .SBTTL PRINT SYMBOL TABLE IN ALPHABETICAL ORDER
023F 396
023F 397
023F 398 PRINT_SYM TABLE:
023F 399      POSHL R10 ;SAVE INT. BUFFER POINTER
03 6B 09  E0 0241 400      BBS #FLGSV_LSTXST,(R11),10$ ;BRANCH IF THERE IS LISTING FILE
      0425 31 0245 401      BRW PRT_ERR_SUM ;NO--GO SEE ABOUT ERROR SUMMARY
0000'CF 01 9F 0248 402 10$: PUSHAB W^MAC$GQ_RNT_P2 ;STACK TIMING BLOCK ADDRESS
      0000'CF 01 9F 024C 403      CALLS #1,W^MAC$TIMER_OFF ;STOP TIMING PASS 2 NOW
0000'CF 01 9F 0251 404      PUSHAB W^MAC$GQ_RNT_SYO ;STACK TIMING BLOCK ADDRESS
0000'CF 01 9A 0255 405      CALLS #1,W^MAC$TIMER_ON ;BEGIN TIMING SYMBOL TABLE OUTPUT
      0000'CF 01 9A 025A 406      MOVZBL #1,W^MAC$GL_LIST_LVL ;FORCE SYMBOL TABLE TO LIST
50 00000000'EF D4 025F 407      CLRL R7 ;RESET STACK POINTER
      51 80 9E 0261 408      MOVAB L^MAC$AB_STB_MSG,R0 ;POINT TO 'SYMBOL TABLE' STRING
0048 8F 20 60 51 2C 0268 409      MOVZBL (R0)+,R1 ;GET ITS LENGTH
      00000000'EF 2C 026B 410      MOVCS R1,(R0),#^A/ /, - ; Copy into subtitle buffer
0272
0277 411      #LST$K_TITLE_SIZE+SYMSK_MAXLEN+1, -
0277 412      L^MAC$AB_SBT_IDNT ;(OVER IDENT AND SUBTITLE)
5A 0000'CF D4 0277 413      CLRL W^MAC$GL_DIRFLG ;CLEAR NEW PAGE OUTPUT FLAG
      0000'CF D0 027B 414      MOVL W^MAC$GL_LINK_PTR+SYMSL_LINK,R10 ; Point to symbol list
      03 12 028C 415      BNEQ 20$ ;IF NEQ GO PRINT SYMBOLS
      00C7 31 0282 416      BRW PRT_SYM_END ;ELSE ALL DONE HERE
      0000'CF D6 0285 417 20$: INCL W^MAC$GL_DIRFLG ;SET NEW PAGE OUTPUT FLAG
      0289 418 PRT_SYM_LOOP:
55 58 5A D0 0289 419      BSBW MAC$LST_PAG_HDR ;SKIP TO NEW PAGE
      0000'CF D0 028C 420      MOVL R10,R8 ;POINT TO SYMBOL
      58 58 D0 028F 421      MOVL W^MAC$GL_LN_PAGE,R5 ; Lines of symbols/page
      58 D5 0294 422 10$: TSTL R8 ;END OF LIST?
      14 13 0296 423      BEQL 20$ ;IF EQL YES
      55 D5 0298 424      TSTL R5 ;NO--END OF PAGE?
      10 15 029A 425      BLEQ 20$ ;IF LEQ YES
05 09 A8 0E E1 029C 426      BBC #SYMSV_SUPR, - ;BRANCH IF NOT SUPPRESSED SYMBOL
      58 68 D0 02A1 427      SYMSW_FLAG(R8),15$
      EE 11 02A4 428      MOVL SYMSL_LINK(R8),R8 ;LINK TO NEXT SYMBOL
      58 68 D0 02A6 429      BRB 10$ ;TRY AGAIN
      E8 55 F4 02A9 430 15$: MOVL SYMSL_LINK(R8),R8 ;NOT SUPPRESSED, LINK TO NEXT SYMBOL
55 0000'CF D0 02AC 431      SOBGEQ R5,10$ ;COUNT AND LOOP
      1B 6B 2A E0 02B1 432 20$: MOVL W^MAC$GL_LN_PAGE,R5 ; Lines of symbols/page
      59 58 D0 02B5 433      BBS #FLGSV_SYM2COL,(R11),40$ ; Branch if two column listing
      02B8 434      MOVL R8,R9 ;R8 POINTS TO FIRST SYMBOL IN
      02B8 435 ;SECOND COLUMN...NOW FIND FIRST
      02B8 436 ;FOR THIRD COLUMN
      59 D5 02B8 437 30$: TSTL R9 ;END OF LIST?
      14 13 02BA 438      BEQL 40$ ;IF EQL YES
      55 D5 02BC 439      TSTL R5 ;NO--END OF PAGE?
      10 15 02BE 440      BLEQ 40$ ;IF LEQ YES
05 09 A9 0E E1 02C0 441      BBC #SYMSV_SUPR, - ;BRANCH IF NOT SUPPRESSED SYMBOL
      59 69 D0 02C5 442      SYMSW_FLAG(R9),35$
      EE 11 02C8 443      MOVL SYMSL_LINK(R9),R9 ;LINK TO NEXT SYMBOL
      59 69 D0 02CA 444      BRB 30$ ;TRY AGAIN
      E8 55 F4 02CD 445 35$: MOVL SYMSL_LINK(R9),R9 ;NOT SUPPRESSED, LINK TO NEXT SYMBOL
      02D0 446      SOBGEQ R5,30$ ;COUNT AND LOOP
      02D0 447 ;
      02D0 448 ; NOW R10 POINTS TO FIRST SYMBOL OF FIRST COLUMN
      02D0 449 ; R8 POINTS TO FIRST SYMBOL OF SECOND COLUMN
      02D0 450 ; R9 POINTS TO FIRST SYMBOL OF THIRD COLUMN
      02D0 451 ;

```

```

0000'CF DD 02D0 452 40$: PUSHL W^MAC$GL_LN_PAGE ; Stack symbols/page
07DC 30 02D4 453 50$: BSBW MAC$LIST_INIT ; INIT LISTING BUFFER AND POINTER
56 5A DO 02D7 454 53$: MOVL R10,R6 ; POINT TO FIRST COLUMN SYMBOL
10 13 02DA 455 BEQL 60$ ; IF EQL THERE IS NONE
05 09 AA OE E1 02DC 456 BBC #SYMSV_SUPR, - ; BRANCH IF NOT SUPPRESSED SYMBOL
02E1 457 SYMSW_FLAG(R10),57$
5A 6A DO 02E1 458 MOVL SYMSL_LINK(R10),R10 ; LINK TO NEXT SYMBOL
F1 11 02E4 459 BRB 53$ ; TRY AGAIN
5A 6A DO 02E6 460 57$: MOVL SYMSL_LINK(R10),R10 ; NOT SUPPRESSED, LINK TO NEXT SYMBOL
06E1 30 02E9 461 BSBW MAC$PRT_SYM_INF ; PRINT THE SYMBOL INFORMATION
56 58 DO 02EC 462 60$: MOVL R8,R6 ; POINT TO SECOND COLUMN SYMBOL
10 13 02EF 463 BEQL 70$ ; IF EQL THERE IS NONE
05 09 A8 OE E1 02F1 464 BBC #SYMSV_SUPR, - ; BRANCH IF NOT SUPPRESSED SYMBOL
02F6 465 SYMSW_FLAG(R8),65$
58 68 DO 02F6 466 MOVL SYMSL_LINK(R8),R8 ; LINK TO NEXT SYMBOL
F1 11 02F9 467 BRB 60$ ; TRY AGAIN
58 68 DO 02FB 468 65$: MOVL SYMSL_LINK(R8),R8 ; NOT SUPPRESSED, LINK TO NEXT SYMBOL
06CC 30 02FE 469 BSBW MAC$PRT_SYM_INF ; PRINT SYMBOL INFORMATION
15 68 2A E0 0301 470 70$: BBS #FLG$V_SYM2COL,(R11),80$ ; Branch if two column listing
56 59 DO 0305 471 73$: MOVL R9,R6 ; POINT TO THIRD COLUMN SYMBOL
10 13 C308 472 BEQL 80$ ; IF EQL THERE IS NONE
05 09 A9 OE E1 030A 473 BBC #SYMSV_SUPR, - ; BRANCH IF NOT SUPPRESSED SYMBOL
030F 474 SYMSW_FLAG(R9),77$
59 69 DO 030F 475 MOVL SYMSL_LINK(R9),R9 ; LINK TO NEXT SYMBOL
F1 11 0312 476 BRB 73$ ; TRY AGAIN
59 69 DO 0314 477 77$: MOVL SYMSL_LINK(R9),R9 ; NOT SUPPRESSED, LINK TO NEXT SYMBOL
06B3 30 0317 478 BSBW MAC$PRT_SYM_INF ; PRINT SYMBOL INFORMATION
00000000'8F 00000000'EF D1 031A 479 80$: CML L^MAC$GL_LIST_PTR,#MAC$AB_LST_END ; LINE THERE?
11 1B 0325 480 BLEQU 90$ ; IF LEQ NO
00000000'8F C3 0327 481 SUBL3 #MAC$AB_LINEBF, - ; YES--FIGURE LINE LENGTH
00000000'EF 032D 482 L^MAC$GL_LIST_PTR, -
0000'CF 0332 483 W^MAC$GL_LINECN
FCCB' 30 0335 484 BSBW MAC$WRTLST ; WRITE IT OUT
99 6E F5 0338 485 90$: SOBGTR (SP),50$ ; LOOP FOR A PAGE
8E D5 033B 486 TSTL (SP)+ ; CLEAN STACK
03 68 2A E1 033D 487 BBC #FLG$V_SYM2COL,(R11),95$ ; Branch if not two column listing
59 58 DO 0341 488 MOVL R8,R9 ; Get next symbol to print
0344 489 95$:
5A 59 DO 0344 490 MOVL R9,R10 ; POINT FOR NEXT PAGE
03 13 0347 491 BEQL 100$ ; IF EQL ALL DONE
FF3D 31 0349 492 BRW PRT_SYM_LOOP ; ELSE CONTINUE
034C 493 100$:
0000'CF 9F 034C 494 PRT_SYM_END:
0000'CF 01 FB 0350 495 PUSHAB W^MAC$GQ_RNT_SYO ; STACK TIMING BLOCK ADDRESS
496 CALLS #1,W^MAC$TIMER_OFF ; FINISH TIMING SYMBOL TABLE OUTPUT

```

```

0355 498 .SBTTL PRINT PSECT SYNOPSIS
0355 499
0355 500 PRT_PSECT SYNOP:
0000'CF 01 9F 0355 501 POSHAB W^MAC$GQ RNT PSY ;STACK TIMING BLOCK ADDRESS
0000'CF 01 FB 0359 502 CALLS #1,W^MAC$TIMER ON ;START TIMING PSECT SYNOPSIS
0000'CF 04 D5 035E 503 TSTL W^MAC$GL_DIRFLG ;WAS SYMBOL TABLE OUTPUT?
0000'CF 04 12 0362 504 BNEQ 10$ ;IF NEQ YES
0000'CF 04 D4 0364 505 CLRL W^MAC$GL_LINE CNT ;NO--ALWAYS OUTPUT NEW PAGE
00000000'GF 9F 0368 506 10$: PUSHAB G^MAC$AB_PSC_RD2A ;CREATE THE HEADER MESSAGE BLOCK
00000000'GF 9F 036E 507 PUSHAB G^MAC$AB_PSC_HD2 ;DYNAMICALLY ON THE STACK.
00000000'GF 00 DD 0374 508 PUSHL #0 ;-- 0 INDICATES A BLANK LINE --
00000000'GF 9F 0376 509 PUSHAB G^MAC$AB_PSC_HDRB ;THIS WILL BE REMOVED LATER, ONCE
00000000'GF 9F 037C 510 PUSHAB G^MAC$AB_PSC_HDRA ;THE NEW SUBTITLE AND SUBSECTION
00000000'GF 9F 0382 511 PUSHAB G^MAC$AB_PSC_HDR ;HAVE BEEN ESTABLISHED.
00000000'GF 00 DD 0388 512 PUSHL #0 ;-- 0 INDICATES A BLANK LINE --
00000000'GF 07 DD 038A 513 PUSHL #7 ;NUMBER OF MESSAGE LINES.
00000000'GF 05 DD 038C 514 PUSHL SP ;ADDRESS OF MESSAGE BLOCK.
00000000'GF 05 DD 038E 515 PUSHL #5 ;STACK # OF LINES WE WILL USE
6E 0000'CF C0 0390 516 ADDL2 W^MAC$GL_PSC_MAX,(SP) ;ADD TO GET TOTAL WE NEED
00000000'EF 9F 0395 517 PUSHAB L^MAC$AB_PSS_MSG ;STACK SUBTITLE STRING
0813'CF 03 FB 0398 518 CALLS #3,W^NEW_SBT_CK_PAGE ;DO NEW PAGE IF NEEDED AND OUTPUT HEADER
0813'CF 50 8E D0 03A0 519 MOVL (SP)+,R0 ;CLEAR THE MESSAGE BLOCK FROM THE STACK.
5E 6E40 DE 03A3 520 MOVAL (SP)[R0],SP ;RESTORE THE STACK POINTER.
56 0000'CF 9E 03A7 521 MOVAB W^PSECT$MAIN,R6 ;POINT TO ABS PSECT
053B 30 03AC 522 BSBW MAC$PSECT_PRINT ;PRINT ITS INFO
56 0000'CF 9E 03AF 523 MOVAB W^PSECT$BLANK,R6 ;POINT TO BLANK PSECT
03 09 A6 07 E1 03B4 524 BBC #SYMSV REF, SYMSW_FLAG(R6),15$ ;HAS IT BEEN REFERENCED?
052E 30 03B9 525 BSBW MAC$PSECT_PRINT ;YES, PRINT ITS INFO
56 0000'CF D0 03BC 526 15$: MOVL W^MAC$GL_PSC_LIST,R6 ;GET PSECT LIST POINTER
08 13 03C1 527 BEQL 30$ ;IF EQL NO MORE
0524 30 03C3 528 20$: BSBW MAC$PSECT_PRINT ;PRINT THIS PSECT
56 66 D0 03C6 529 MOVL (R6),R6 ;NEXT PSECT
F8 12 03C9 530 BNEQ 20$ ;IF NEQ THERE IS A NEXT
0000'CF 9F 03CB 531 30$: PUSHAB W^MAC$GQ RNT PSY ;STACK TIMING BLOCK ADDRESS
0000'CF 01 FB 03CF 532 CALLS #1,W^MAC$TIMER_OFF ;STOP TIMING PSECT SYNOPSIS

```

```

03D4 534 .SBTTL PRINT CROSS-REFERENCE IF REQUESTED
03D4 535
03D4 536 PRT_CROSS_REF:
0000'CF D5 03D4 537 TSTL W^MAC$GL_CRF_FLG ;WAS THERE ANY CROSS REFERENCE?
6E 13 03D8 538 BEQL 40$ ;IF EQL NO--SKIP IT ALL
0000'CF 9F 03DA 539 PUSHAB W^MAC$GQ_RNT_CRF ;STACK TIMING BLOCK ADDRESS
0000'CF 01 FB 03DE 540 CALLS #1,W^MAC$TIMER_ON ;START TIMING CREF PHASE
59 D4 03E3 541 CLRL R9 ;CLEAR INDEX INTO TABLES
50 00000000'EF49 D0 03E5 542 10$: MOVL L^MAC$AL_CRF_TB5[R9],R0 ;ARE WE DONE?
50 13 03ED 543 BEQL 30$ ;IF EQL YES
51 00000000'EF49 9A 03EF 544 MOVZBL L^MAC$AB_CRF_TB6[R9],R1 ;GET BIT # TO CHECK
3E 0000'CF 51 E1 03F7 545 BBC R1,W^MAC$GL_CRF_FLG,20$ ;IF CLEAR TRY NEXT CREF AREA
60 D5 03FD 546 TSTL (R0) ;DID WE CREF ANYTHING?
3A 13 03FF 547 BEQL 20$ ;IF EQL NO--DO NEXT
50 59 D0 0401 548 MOVL R9, R0 ; Pass index.
51 D4 0404 549 CLRL R1 ; Assume normal cref width.
52 00000000'EF49 D0 0406 550 MOVL L^MAC$AL_CRF_TB3[R9],R2 ;GET CONTROL TABLE ADDRESS
53 00000000'EF49 D0 040E 551 MOVL L^MAC$AL_CRF_TB7[R9],R3 ; Get column width flag
18 13 0416 552 BEQL 15$ ; If EQL not a variable field
48 A2 00000000'EF49 D0 0418 553 MOVL L^MAC$AL_CRF_TB8[R9], - ; Set KEY1FIELD entry
0B 6B 53 E1 0421 554 CRF$K1FMTB[R2] ; assuming narrow field
51 D6 0425 555 INCL R3,(RT1),15$ ; Branch if narrow column listing
48 A2 00000000'EF49 D0 0427 556 MOVL R1 ; Set wide listing flag.
0430 557 MOVL L^MAC$AL_CRF_TB9[R9], - ; Set KEY1FIELD ENTRY
0430 558 CRF$K1FMTB[R2] ; for wide field
53 00000000'EF49 9A 0430 559 15$: MOVZBL L^MAC$AB_CRF_TB4[R9],R3 ;GET DEFS_REFS OR VALS_REFS
0354 30 0438 561 BSBW CREF_TREE_OUT ;DO THIS CREF
59 D6 043B 562 20$: INCL R9 ;NEXT CREF
A6 11 043D 563 BRB 10$ ;DO IT
0000'CF 9F 043F 564 30$: PUSHAB W^MAC$GQ_RNT_CRF ;STACK TIMING BLOCK ADDRESS
0000'CF 01 FB 0443 565 CALLS #1,W^MAC$TIMER_OFF ;STOP TIMING CROSS REFERENCE
0000'CF 9F 0448 566 40$: PUSHAB W^MAC$GQ_RNT_TOT ;STACK TIMING BLOCK ADDRESS
0000'CF 01 FB 044C 567 CALLS #1,W^MAC$TIMER_OFF ;STOP TIMING OF ASSEMBLER NOW

```



```

0752 786 .SBTTL PRINT COMMAND LINE
0752 787
0752 788 PRT_CMD_LIN:
0752 789 BSBW MAC$WRT_BLNKLN ;SKIP A LINE
00000000'EF DD 0755 790 PUSHL L^MAC$GL_CMDLIN ;STACK COMMAND LINE ADDRESS
00000000'EF DD 075B 791 PUSHL L^MAC$GL_CMDLEN ;AND ITS LENGTH
50 00000000'EF 9E 0761 792 MOVAB L^MAC$AB_CMD_FAO,RO ;POINT TO FAO CONTROL STRING
F895' 30 0768 793 BSBW MAC$FAOUTS ;FORMAT THE STRING
50 00000000'EF C1 076B 794 ADDL3 L^MAC$GL_ERRCT,L^MAC$GL_WARNCT,RO ;ERRORS OR WARNINGS?
50 0000'CF C0 0777 795 ADDL2 W^MAC$GL_INFOCNT,RO ; or information messages?
03 13 077C 796 BEQL 10$ ;IF EQL NO
F87F' 30 077E 797 BSBW MAC$WRITE_TERM ;YES--WRITE TO TERMINAL ALSO
F87C' 30 0781 798 10$: BSBW MAC$WRTLST ;WRITE TO LISTING FILE
SE 08 C0 0784 799 ADDL2 #2*4,SP ;CLEAR 2 LW FROM STACK
0787 800 FIN_ASM_EXIT:
0787 801 POPL R10 ;RESTORE R10
0380 8F BA 078A 802 POPR #*M<R7,R8,R9> ;RESTORE REGISTERS
05 078E 803 RSB ;**EXIT FROM MAC$FINISH_ASM

```

MAC
Pse

PSE
-77
.
SAE
MAC
MAC

Pha

In
Com
Pas
Syn
Pas
Syn
Pse
Cra
101

The
888
The
110
43

Mac

-S
-S
TO
130
The
MAC

```

078F 805          .SBTTL OUTPUT ONE PHASE OF CROSS REFERENCE
078F 806
078F 807 :++
078F 808 : FUNCTIONAL DESCRIPTION:
078F 809 :
078F 810 :         THIS ROUTINE OUTPUTS ONE PHASE OF THE CROSS REFERENCE.
078F 811 :
078F 812 : INPUTS:
078F 813 :
078F 814 :         R0      CROSS REFERENCE TYPE INDEX
078F 815 :         R1      0 => NORMAL WIDTH LISTING, 1 => WIDE LISTING FORMAT
078F 816 :         R2      POINTER TO CREF CONTROL TABLE
078F 817 :         R3      CRF$K_VALS_REFS OR CRF$K_DEFS_REFS
078F 818 :
078F 819 :--
078F 820
078F 821 CREF_TREE OUT:
00000000'EF D4 078F 822          CRL      L^MAC$GL_LINE_CNT      : ALWAYS FORCE NEW PAGE FOR CREF
          53 DD 0795 823          PUSHL   R3                : SAVE REFS OR DEFS_REFS FLAG
          52 DD 0797 824          PUSHL   R2                : SAVE CREF CONTROL_BLOCK ADDRESS
          10 51 E9 0799 825          BLBC    R1, 10$          : CHECK THE LISTING CONTROL INDICATOR.
00000000'GF40 DD 079C 826          PUSHL   G^MAC$AL_CRF_TB2WA[R0] : WIDE LISTING FORMAT HAS BEEN CHOSEN.
00000000'GF40 DD 07A3 827          PUSHL   G^MAC$AL_CRF_TB2W [R0] : STACK THE SECONDARY HEADER LINES.
          0E 11 07AA 828          BRB     20$              : JOIN COMMON CODE FOR SECTION HEADING.
00000000'GF40 DD 07AC 829 10$:    PUSHL   G^MAC$AL_CRF_TB2A [R0] : NORMAL WIDTH LISTING HAS BEEN CHOSEN.
00000000'GF40 DD 07B3 830          PUSHL   G^MAC$AL_CRF_TB2 [R0]  : STACK THE SECONDARY HEADER LINES.
          00 DD 07BA 831 20$:    PUSHL   #0                : -- 0 INDICATES A BLANK LINE --
00000000'GF40 DD 07BC 832          PUSHL   G^MAC$AL_CRF_TB1B [R0] : STACK THE SECTION HEADER LINES...
00000000'GF40 DD 07C3 833          PUSHL   G^MAC$AL_CRF_TB1A [R0] : (THE ONES THAT FORM THE BOXED IN
00000000'GF40 DD 07CA 834          PUSHL   G^MAC$AL_CRF_TB1 [R0]  : SECTION HEADERS).
          00 DD 07D1 835          PUSHL   #0                : -- 0 INDICATES A BLANK LINE --
          07 DD 07D3 836          PUSHL   #7                : NUMBER OF HEADER LINES IN THIS BLOCK.
          5E DD 07D5 837          PUSHL   SP                : HEADER MESSAGE BLOCK ADDRESS.
          09 DD 07D7 838          PUSHL   #5+4             : STACK # LINES ACTUALLY NEEDED
00000000'GF 9F 07D9 839          PUSHAB  G^MAC$AB_CRF_MSG      : STACK LINE 2 HEADER MESSAGE
0813'CF 03 FB 07DF 840          CALLS   #3,W^NEW_SBT_CK_PAGE : DO HEADER THING
          50 8E D0 07E4 841          MOVL   (SP)+, R0        : CLEAR THE MESSAGE BLOCK FROM THE STACK.
          5E 6E40 DE 07E7 842          MOVAL  (SP)[R0], SP    : RESTORE THE STACK POINTER.
          50 8ED0 07EB 843          POPL   R0                : GET CREF CONTROL BLOCK ADDRESS
          51 8ED0 07EE 844          POPL   R1                : GET THE REFS OR DEFS_REFS FLAG
00000000'8F DD 07F1 845          PUSHL   #CRF$K_DELETE      : DELETE TREE FLAG FOR CREF
          51 DD 07F7 846          PUSHL   R1                : SET REFS OR DEFS_REFS FLAG ON STACK
          0000'CF DD 07F9 847          PUSHL   W^MAC$GL_LN_PAGE  : # LINES ON SUBSEQUENT PAGES
00000000'EF DD 07FD 848          PUSHL   L^MAC$GL_LINE_CNT  : # LINES ON FIRST PAGE
00000084 8F DD 0803 849          PUSHL   #132             : WIDTH OF A LINE
          50 DD 0809 850          PUSHL   R0                : CREF CONTROL BLOCK ADDRESS
00000000'GF 06 FB 080B 851          CALLS   #6,G^CRF$OUT    : OUTPUT CROSS REFERENCE
          05 0812 852          RSB

```

```

0813 854 .SBTTL SET UP FOR NEW SUBTITLE AND CHECK NEW PAGE
0813 855
0813 856 :++
0813 857 : FUNCTIONAL DESCRIPTION:
0813 858 :
0813 859 : THIS ROUTINE PUTS A NEW LINE IN THE SUBTITLE BUFFER FOR
0813 860 : PSECT SYNOPSIS AND THE RUN STATISTICS.
0813 861 :
0813 862 : INPUTS:
0813 863 :
0813 864 : 4(AP) ADDRESS OF STRING TO PUT IN TITLE BUFFER (COUNT,TEXT)
0813 865 : 8(AP) # LINES NEEDED LEFT ON PAGE
0813 866 : 12(AP) ADDRESS OF HEADER MESSAGE BLOCK
0813 867 :
0813 868 : NOTE: THE FORMAT OF THE HEADER MESSAGE BLOCK IS THE SAME AS
0813 869 : A STANDARD VAX/VMS ARGUMENT LIST, WHOSE ENTRIES ARE THE
0813 870 : ADDRESSES OF FAO CONTROL STRING DESCRIPTORS.
0813 871 :--
0813 872
0813 873
0813 874 .ENTRY NEW_SBT_CHK_PAGE, ^M<R2> ;ENTRY POINT
50 04 AC 0004 0815 875 MOVL 4(AP),R0 ;GET SUBTITLE STRING ADDRESS
51 80 9A 0819 876 MOVZBL (R0)+,R1 ;GET ITS LENGTH
20 60 51 2C 081C 877 MOVCS R1,(R0),#^A/ /,- ;COPY OVER SUBTITLE AND IDENT
00000000'EF 0048 8F 0820 878 #LSTSK_TITLE_SIZ+SYMSK_MAXLEN+1,-
0828 879 L^MAC$AB_SBT_IDNT
50 00000000'EF 08 AC C3 0828 880 SUBL3 8(AP),L^MAC$GL_LINE_CNT,R0 ;SEE IF ROOM LEFT ON PAGE
03 14 0831 881 BGTR 10$ ;IF GTR YES
F7CA' 30 0833 882 BSBW MAC$LST_PAG_HDR ;NO--MAKE NEW PAGE NOW
52 0C AC D0 0836 883 10$: MOVL 12(AP),R2 ;GET ADDRESS OF HEADER MESSAGE BLOCK
51 82 D0 083A 884 MOVL (R2)+,R1 ;GET THE NUMBER OF HEADER MESSAGE LINES
14 13 083D 885 BEQL 50$ ;ANY PROVIDED? RETURN IF NOT.
50 82 D0 083F 886 20$: MOVL (R2)+,R0 ;GET THE NEXT HEADER LINE
06 BB 0842 887 PUSHR #^M<R1,R2> ;PRESERVE REGISTERS ACROSS CALLS
05 13 0844 888 BEQL 30$ ;BLANK LINE?
F7B7' 30 0846 889 BSBW MAC$WRT_FAOUTS ;NO, FORMAT AND OUTPUT THE STRING
03 11 0849 890 BRB 40$ ;CHECK FOR MORE HEADER LINES...
F7B2' 30 084B 891 30$: BSBW MAC$WRT_BLNKLIN ;SKIP A LINE.
06 BA 084E 892 40$: POPR #^M<R1,R2> ;RESTORE REGISTERS.
EC 51 F5 0850 893 SOBGTR R1, 20$ ;ANY MORE HEADER LINES?
04 0853 894 50$: RET ;ALL DONE

```

```

0854 896      .SBTTL OUTPUT SYMBOL NAME AND VALUE FOR DEBUG
0854 897
0854 898      ;++
0854 899      ; FUNCTIONAL DESCRIPTION:
0854 900      ;
0854 901      ;     THIS ROUTINE OUTPUTS THE NAME OF A SYMBOL FOR THE DEBUGGER.
0854 902      ;
0854 903      ; INPUTS:
0854 904      ;
0854 905      ;     R6     POINTS TO SYMBOL BLOCK
0854 906      ;
0854 907      ;--
0854 908
0854 909 MAC$DBG_NAM_OUT:
54 04 A6 9A 0854 910      MOVZBL  SYM$B_NAME(R6),R4      ; Get offset to name/count (which is
0858 911      ; also total size of count and name
55 56 54 C3 0858 912      SUBL3   R4,R6,R5      ; Set pointer to count/name
50 85 90 085C 913 10$:  MOVB    (R5)+,R0      ; GET CHARACTER OF NAME
      F79E' 30 085F 914      BSBW    MAC$STOIM
      F7 54 F5 0862 915      SOBGTR  R4,10$      ; LOOP FOR WHOLE NAME
0865 916      RSB
0866 917
0866 918      ;++
0866 919      ; FUNCTIONAL DESCRIPTION:
0866 920      ;
0866 921      ;     THIS ROUTINE OUTPUTS THE VALUE OF A SYMBOL FOR THE DEBUGGER.
0866 922      ;
0866 923      ; INPUTS:
0866 924      ;
0866 925      ;     R6     POINTS TO SYMBOL BLOCK
0866 926      ;
0866 927      ;--
0866 928
0866 929 MAC$DBG_VAL_OUT:
7E 0C A6 9A 0866 930      $OBJ_CHKBYT #TIR$C_STA_PL      ; STACK PSECT PLUS LONGWORD
086C 931      MOVZBL  SYM$B_SEG(R6),--(L,P)      ; GET THE SEGMENT NUMBER.
0870 932      BEQL    10$      ; DON'T FIDDLE WITH THE ABS PSECT!
0872 933      BBS     #SYM$V_REF,-      ; HAS THE BLANK PSECT BEEN REF'D?
02 0000009'GF E0 0874 934      G^PSECT$BLANK+SYM$W_FLAG,10$ ; IF NOT, IT'LL BE REMOVED, SO DECR
      6E D7 087A 935      DECL    (SP)      ; THE SEG# TO PRESERVE PSECT ALIGNMENT.
      8E D5 087C 936 10$:  $OBJ_OUTBYT (SP)      ; EMIT THE SYMBOL'S SEGMENT #
55 05 A6 9E 0882 937      TSTL    (SP)+      ; CLEAN UP THE STACK
      F775' 30 0884 938      MOVAB   SYM$L_VAL(R6),R5      ; POINT TO VALUE
0888 939      BSBW    MAC$OIT_LW      ; OUTPUT VALUE
088B 940      $OBJ_CHKBYT #TIR$C_STO_LW      ; STORE LONGWORD
0891 941      RSB

```

```

0892 943      .SBTTL  OUTPUT PSECT RECORDS TO DEBUG
0892 944
0892 945      :++
0892 946      : FUNCTIONAL DESCRIPTION:
0892 947      :
0892 948      : THIS ROUTINE OUTPUTS PSECT INFORMATION TO THE DEBUGGER
0892 949      :
0892 950      : INPUTS:
0892 951      :
0892 952      : R6      POINTS TO SYMBOL BLOCK FOR PSECT
0892 953      :
0892 954      :--
0892 955
0892 956 MAC$DBG_PSECT:
50 04 A6 9A 0892 957 MOVZBL SYMSB_NAME(R6),R0      ; Get offset to symbol count/name
50 56 50 C3 0896 958 SUBL3  RO,R6,R0      ; Form address of count/name
50 0B 60 81 089A 959 ADDB3  (R0),#DBG$K_PSECT_LEN-1,R0 ; Figure size
      F75F' 30 089E 960 BSBW  MAC$STOIM      ;STORE IT
50 0B 8F 9A 08A1 961 MOVZBL #DBG$C_PSECT,R0      ;PSECT TYPE
      F758' 30 08A5 962 BSBW  MAC$STOIM
      50 D4 08A8 963 CLRL  RO      ;MBZ BYTE
      F753' 30 08AA 964 BSBW  MAC$STOIM
      08AD 965 $OBJ_CHKBYT #TIR$C_STA_PB ;STACK PSECT BASE PLUS BYTE OFFSET
7E 0C A6 9A 08B3 966 MOVZBL SYMSB_SEG(R6),--(SP) ;GET THE SEGMENT NUMBER.
      0A 13 08B7 967 BEQL  5$      ;DON'T FIDDLE WITH THE ABS PSECT!
      07 E0 08B9 968 BBS   #SYMSV_REF,- ;HAS THE BLANK PSECT BEEN REF'D?
02 00000009'GF 08BB 969      G^PSECT$BLANK+SYMSW_FLAG,5$ ;IF NOT, IT'LL BE REMOVED, SO DECR
      6E D7 08C1 970 DECL  (SP) ;THE SEG# TO PRESERVE PSECT ALIGNMENT.
      8E D5 08C3 971 5$: $OBJ_OUTBYT (SP) ;EMIT THE SYMBOL'S SEGMENT #
      50 D4 08C9 972 TSTL  (SP)+ ;AND CLEAN UP THE STACK.
      F730' 30 08CB 973 CLRL  RO ;USE OFFSET OF 0
      08CD 974 BSBW  MAC$OUTOBJ
      08D0 975 $OBJ_CHKBYT #TIR$C_STO_PDR ;STORE POSITION INDEPENDENT DATA REFERENCE.
55 05 A6 9E 08D6 976 BSBW  MAC$DBG_NAM_OUT ;SEND PSECT NAME
      54 04 9A 08D9 977 MOVAB  SYMSL_VAL(R6),R5 ;POINT TO LENGTH
      50 85 90 08DD 978 MOVZBL #4,R4 ;LOOP COUNT
      F71A' 30 08E0 979 10$: MOVB  (R5)+,RO ;GET BYTE OF VALUE
      F7 54 F5 08E3 980 BSBW  MAC$STOIM
      05 08E6 981 SOBGTR R4,10$
      08E9 982 RSB

```

```

08EA 984      .SBTTL PRINT PSECT INFORMATION FOR 1 PSECT
08EA 985
08EA 986 :++
08EA 987 : FUNCTIONAL DESCRIPTION:
08EA 988 :
08EA 989 :     THIS ROUTINE PRINTS THE PSECT SYNOPSIS FOR ONE PSECT.
08EA 990 :
08EA 991 : INPUTS:
08EA 992 :
08EA 993 :     R6      PSECT BLOCK ADDRESS
08EA 994 :
08EA 995 :--
08EA 996
08EA 997 MAC$PSECT PRINT:
08EA 998      BSBW  MAC$LIST_INIT          ;INIT LISTING BUFFER
50 0C A6 9A 08ED 999      MOVZBL PSC$B_SEG(R6),R0      ;GET SEGMENT NUMBER
08EA 1000      BEQL  2$                    ;DON'T FIDDLE WITH THE ABS PSECT.
08EA 1001      BBS   #SYMSV_REF,-        ;HAS THE BLANK PSECT BEEN REF'D?
02 00000009'GF 08F5 1002      G^PSECT$BLANK+SYMSW_FLAG,2$ ;IF NOT, IT'LL BE REMOVED, SO DECR
08EA 1003      DECL  R0                    ;THE SEG# TO PRESERVE PSECT ALIGNMENT.
08EA 1004 2$:  PUSHL  R0                    ;STACK IT
08EA 1005      PUSHL  R0                    ;TWICE FOR FAO
05 A6 DD 0901 1006      PUSHL  PSC$L_MAXLGTH(R6)      ;STACK THE MAX LENGTH
08EA 1007      PUSHL  (SP)                  ;COPY IT (HEX AND DEC OUTPUT)
08EA 1008      BGTR  5$                    ;BRANCH IF POSITIVE LENGTH
08EA 1009      CLRL  4(SP)                  ;ELSE PRINT DECIMAL AS 0
08EA 1010 5$:  MOVZBL PSC$B_NAME(R6),R0      ; Get offset to symbol count/name
08EA 1011      SUBL3  R0,R6,-(SP)          ; and put its address on stack
08EA 1012      MOVAB  L^MAC$AB_PSS_FAO,R0  ;POINT TO FAO CONTROL STRING
08EA 1013      BSBW  MAC$FAOUTS          ;FORMAT STRING
08EA 1014      ADDL2  #5*4,SP              ;CLEAR THE STACK
50 00000000'EF 08F1 1015      ADDL3  #MAC$K_LIST_SIZE,L^MAC$GL_LIST_PTR ;FIGURE LENGTH OF LINE SO FAR
08EA 1016      MOVAB  L^MAC$AB_LST_END(R0),- ;INIT LISTING POINTER
08EA 1017      MOVZWL PSC$W_OPTIONS(R6),R5 ;GET PSECT OPTIONS
08EA 1018      MOVL  #1,R4                    ;FIRST BIT
08EA 1019      PUSHL #PSC$K_NO_OPTNS      ; Loop once for each option
08EA 1020
08EA 1021 :
08EA 1022 : LOOP, PRINTING PSECT OPTIONS
08EA 1023 :
08EA 1024 10$: BSBW  PSECT_OPT_MATCH      ;PRINT THIS OPTION
08EA 1025      ADDL2  R4,R4                    ;NEXT BIT
08EA 1026      SOBGTR (SP),10$              ;LOOP FOR ALL
08EA 1027      TSTL  (SP)+                  ;CLEAN STACK
08EA 1028      MOVZWL PSC$W_OPTIONS(R6),R4 ;COPY OPTIONS AGAIN
54 000003FF 8F CA 094F 1029      BICL2  #^XPSC$M_ALLOPTNS,R4      ; Trim all but alignment
08EA 1030      BISW2  #PSC$M_ALIGNFLG,R4    ;SET THE ALIGNMENT FLAG BIT
08EA 1031      MOVL  R4,R5                    ;POSITIVE SENSE
08EA 1032      BSBW  PSECT_OPT_MATCH      ;PRINT ALIGNMENT
08EA 1033      BLBS  R0,20$                  ;BRANCH IF FOUND
08EA 1034      ASHL  #-PSC$V_ALIGNMENT,R4,R0 ;GET ALIGNMENT
08EA 1035      BSBW  MAC$DEC_OUT_L2R        ;OUTPUT ALIGNMENT
00000000'EF 00000000'8F C3 096C 1036 20$:  SUBL3  #MAC$AB_LINEBF,L^MAC$GL_LIST_PTR,- ;FIGURE LINE LENGTH
08EA 1037      MOVZWL L^MAC$GL_LIST_PTR,-
08EA 1038      BRW   MAC$WRTLST          ;WRITE LINE TO LISTING AND RETURN
08EA 1039
08EA 1040
08EA 1041
08EA 1042
08EA 1043
08EA 1044
08EA 1045
08EA 1046
08EA 1047
08EA 1048
08EA 1049
08EA 1050
08EA 1051
08EA 1052
08EA 1053
08EA 1054
08EA 1055
08EA 1056
08EA 1057
08EA 1058
08EA 1059
08EA 1060
08EA 1061
08EA 1062
08EA 1063
08EA 1064
08EA 1065
08EA 1066
08EA 1067
08EA 1068
08EA 1069
08EA 1070
08EA 1071
08EA 1072
08EA 1073
08EA 1074
08EA 1075
08EA 1076
08EA 1077
08EA 1078
08EA 1079
08EA 1080
08EA 1081
08EA 1082
08EA 1083
08EA 1084
08EA 1085
08EA 1086
08EA 1087
08EA 1088
08EA 1089
08EA 1090
08EA 1091
08EA 1092
08EA 1093
08EA 1094
08EA 1095
08EA 1096
08EA 1097
08EA 1098
08EA 1099
08EA 1100
08EA 1101
08EA 1102
08EA 1103
08EA 1104
08EA 1105
08EA 1106
08EA 1107
08EA 1108
08EA 1109
08EA 1110
08EA 1111
08EA 1112
08EA 1113
08EA 1114
08EA 1115
08EA 1116
08EA 1117
08EA 1118
08EA 1119
08EA 1120
08EA 1121
08EA 1122
08EA 1123
08EA 1124
08EA 1125
08EA 1126
08EA 1127
08EA 1128
08EA 1129
08EA 1130
08EA 1131
08EA 1132
08EA 1133
08EA 1134
08EA 1135
08EA 1136
08EA 1137
08EA 1138
08EA 1139
08EA 1140
08EA 1141
08EA 1142
08EA 1143
08EA 1144
08EA 1145
08EA 1146
08EA 1147
08EA 1148
08EA 1149
08EA 1150
08EA 1151
08EA 1152
08EA 1153
08EA 1154
08EA 1155
08EA 1156
08EA 1157
08EA 1158
08EA 1159
08EA 1160
08EA 1161
08EA 1162
08EA 1163
08EA 1164
08EA 1165
08EA 1166
08EA 1167
08EA 1168
08EA 1169
08EA 1170
08EA 1171
08EA 1172
08EA 1173
08EA 1174
08EA 1175
08EA 1176
08EA 1177
08EA 1178
08EA 1179
08EA 1180
08EA 1181
08EA 1182
08EA 1183
08EA 1184
08EA 1185
08EA 1186
08EA 1187
08EA 1188
08EA 1189
08EA 1190
08EA 1191
08EA 1192
08EA 1193
08EA 1194
08EA 1195
08EA 1196
08EA 1197
08EA 1198
08EA 1199
08EA 1200
08EA 1201
08EA 1202
08EA 1203
08EA 1204
08EA 1205
08EA 1206
08EA 1207
08EA 1208
08EA 1209
08EA 1210
08EA 1211
08EA 1212
08EA 1213
08EA 1214
08EA 1215
08EA 1216
08EA 1217
08EA 1218
08EA 1219
08EA 1220
08EA 1221
08EA 1222
08EA 1223
08EA 1224
08EA 1225
08EA 1226
08EA 1227
08EA 1228
08EA 1229
08EA 1230
08EA 1231
08EA 1232
08EA 1233
08EA 1234
08EA 1235
08EA 1236
08EA 1237
08EA 1238
08EA 1239
08EA 1240
08EA 1241
08EA 1242
08EA 1243
08EA 1244
08EA 1245
08EA 1246
08EA 1247
08EA 1248
08EA 1249
08EA 1250
08EA 1251
08EA 1252
08EA 1253
08EA 1254
08EA 1255
08EA 1256
08EA 1257
08EA 1258
08EA 1259
08EA 1260
08EA 1261
08EA 1262
08EA 1263
08EA 1264
08EA 1265
08EA 1266
08EA 1267
08EA 1268
08EA 1269
08EA 1270
08EA 1271
08EA 1272
08EA 1273
08EA 1274
08EA 1275
08EA 1276
08EA 1277
08EA 1278
08EA 1279
08EA 1280
08EA 1281
08EA 1282
08EA 1283
08EA 1284
08EA 1285
08EA 1286
08EA 1287
08EA 1288
08EA 1289
08EA 1290
08EA 1291
08EA 1292
08EA 1293
08EA 1294
08EA 1295
08EA 1296
08EA 1297
08EA 1298
08EA 1299
08EA 1300
08EA 1301
08EA 1302
08EA 1303
08EA 1304
08EA 1305
08EA 1306
08EA 1307
08EA 1308
08EA 1309
08EA 1310
08EA 1311
08EA 1312
08EA 1313
08EA 1314
08EA 1315
08EA 1316
08EA 1317
08EA 1318
08EA 1319
08EA 1320
08EA 1321
08EA 1322
08EA 1323
08EA 1324
08EA 1325
08EA 1326
08EA 1327
08EA 1328
08EA 1329
08EA 1330
08EA 1331
08EA 1332
08EA 1333
08EA 1334
08EA 1335
08EA 1336
08EA 1337
08EA 1338
08EA 1339
08EA 1340
08EA 1341
08EA 1342
08EA 1343
08EA 1344
08EA 1345
08EA 1346
08EA 1347
08EA 1348
08EA 1349
08EA 1350
08EA 1351
08EA 1352
08EA 1353
08EA 1354
08EA 1355
08EA 1356
08EA 1357
08EA 1358
08EA 1359
08EA 1360
08EA 1361
08EA 1362
08EA 1363
08EA 1364
08EA 1365
08EA 1366
08EA 1367
08EA 1368
08EA 1369
08EA 1370
08EA 1371
08EA 1372
08EA 1373
08EA 1374
08EA 1375
08EA 1376
08EA 1377
08EA 1378
08EA 1379
08EA 1380
08EA 1381
08EA 1382
08EA 1383
08EA 1384
08EA 1385
08EA 1386
08EA 1387
08EA 1388
08EA 1389
08EA 1390
08EA 1391
08EA 1392
08EA 1393
08EA 1394
08EA 1395
08EA 1396
08EA 1397
08EA 1398
08EA 1399
08EA 1400
08EA 1401
08EA 1402
08EA 1403
08EA 1404
08EA 1405
08EA 1406
08EA 1407
08EA 1408
08EA 1409
08EA 1410
08EA 1411
08EA 1412
08EA 1413
08EA 1414
08EA 1415
08EA 1416
08EA 1417
08EA 1418
08EA 1419
08EA 1420
08EA 1421
08EA 1422
08EA 1423
08EA 1424
08EA 1425
08EA 1426
08EA 1427
08EA 1428
08EA 1429
08EA 1430
08EA 1431
08EA 1432
08EA 1433
08EA 1434
08EA 1435
08EA 1436
08EA 1437
08EA 1438
08EA 1439
08EA 1440
08EA 1441
08EA 1442
08EA 1443
08EA 1444
08EA 1445
08EA 1446
08EA 1447
08EA 1448
08EA 1449
08EA 1450
08EA 1451
08EA 1452
08EA 1453
08EA 1454
08EA 1455
08EA 1456
08EA 1457
08EA 1458
08EA 1459
08EA 1460
08EA 1461
08EA 1462
08EA 1463
08EA 1464
08EA 1465
08EA 1466
08EA 1467
08EA 1468
08EA 1469
08EA 1470
08EA 1471
08EA 1472
08EA 1473
08EA 1474
08EA 1475
08EA 1476
08EA 1477
08EA 1478
08EA 1479
08EA 1480
08EA 1481
08EA 1482
08EA 1483
08EA 1484
08EA 1485
08EA 1486
08EA 1487
08EA 1488
08EA 1489
08EA 1490
08EA 1491
08EA 1492
08EA 1493
08EA 1494
08EA 1495
08EA 1496
08EA 1497
08EA 1498
08EA 1499
08EA 1500
08EA 1501
08EA 1502
08EA 1503
08EA 1504
08EA 1505
08EA 1506
08EA 1507
08EA 1508
08EA 1509
08EA 1510
08EA 1511
08EA 1512
08EA 1513
08EA 1514
08EA 1515
08EA 1516
08EA 1517
08EA 1518
08EA 1519
08EA 1520
08EA 1521
08EA 1522
08EA 1523
08EA 1524
08EA 1525
08EA 1526
08EA 1527
08EA 1528
08EA 1529
08EA 1530
08EA 1531
08EA 1532
08EA 1533
08EA 1534
08EA 1535
08EA 1536
08EA 1537
08EA 1538
08EA 1539
08EA 1540
08EA 1541
08EA 1542
08EA 1543
08EA 1544
08EA 1545
08EA 1546
08EA 1547
08EA 1548
08EA 1549
08EA 1550
08EA 1551
08EA 1552
08EA 1553
08EA 1554
08EA 1555
08EA 1556
08EA 1557
08EA 1558
08EA 1559
08EA 1560
08EA 1561
08EA 1562
08EA 1563
08EA 1564
08EA 1565
08EA 1566
08EA 1567
08EA 1568
08EA 1569
08EA 1570
08EA 1571
08EA 1572
08EA 1573
08EA 1574
08EA 1575
08EA 1576
08EA 1577
08EA 1578
08EA 1579
08EA 1580
08EA 1581
08EA 1582
08EA 1583
08EA 1584
08EA 1585
08EA 1586
08EA 1587
08EA 1588
08EA 1589
08EA 1590
08EA 1591
08EA 1592
08EA 1593
08EA 1594
08EA 1595
08EA 1596
08EA 1597
08EA 1598
08EA 1599
08EA 1600
08EA 1601
08EA 1602
08EA 1603
08EA 1604
08EA 1605
08EA 1606
08EA 1607
08EA 1608
08EA 1609
08EA 1610
08EA 1611
08EA 1612
08EA 1613
08EA 1614
08EA 1615
08EA 1616
08EA 1617
08EA 1618
08EA 1619
08EA 1620
08EA 1621
08EA 1622
08EA 1623
08EA 1624
08EA 1625
08EA 1626
08EA 1627
08EA 1628
08EA 1629
08EA 1630
08EA 1631
08EA 1632
08EA 1633
08EA 1634
08EA 1635
08EA 1636
08EA 1637
08EA 1638
08EA 1639
08EA 1640
08EA 1641
08EA 1642
08EA 1643
08EA 1644
08EA 1645
08EA 1646
08EA 1647
08EA 1648
08EA 1649
08EA 1650
08EA 1651
08EA 1652
08EA 1653
08EA 1654
08EA 1655
08EA 1656
08EA 1657
08EA 1658
08EA 1659
08EA 1660
08EA 1661
08EA 1662
08EA 1663
08EA 1664
08EA 1665
08EA 1666
08EA 1667
08EA 1668
08EA 1669
08EA 1670
08EA 1671
08EA 1672
08EA 1673
08EA 1674
08EA 1675
08EA 1676
08EA 1677
08EA 1678
08EA 1679
08EA 1680
08EA 1681
08EA 1682
08EA 1683
08EA 1684
08EA 1685
08EA 1686
08EA 1687
08EA 1688
08EA 1689
08EA 1690
08EA 1691
08EA 1692
08EA 1693
08EA 1694
08EA 1695
08EA 1696
08EA 1697
08EA 1698
08EA 1699
08EA 1700
08EA 1701
08EA 1702
08EA 1703
08EA 1704
08EA 1705
08EA 1706
08EA 1707
08EA 1708
08EA 1709
08EA 1710
08EA 1711
08EA 1712
08EA 1713
08EA 1714
08EA 1715
08EA 1716
08EA 1717
08EA 1718
08EA 1719
08EA 1720
08EA 1721
08EA 1722
08EA 1723
08EA 1724
08EA 1725
08EA 1726
08EA 1727
08EA 1728
08EA 1729
08EA 1730
08EA 1731
08EA 1732
08EA 1733
08EA 1734
08EA 1735
08EA 1736
08EA 1737
08EA 1738
08EA 1739
08EA 1740
08EA 1741
08EA 1742
08EA 1743
08EA 1744
08EA 1745
08EA 1746
08EA 1747
08EA 1748
08EA 1749
08EA 1750
08EA 1751
08EA 1752
08EA 1753
08EA 1754
08EA 1755
08EA 1756
08EA 1757
08EA 1758
08EA 1759
08EA 1760
08EA 1761
08EA 1762
08EA 1763
08EA 1764
08EA 1765
08EA 1766
08EA 1767
08EA 1768
08EA 1769
08EA 1770
08EA 1771
08EA 1772
08EA 1773
08EA 1774
08EA 1775
08EA 1776
08EA 1777
08EA 1778
08EA 1779
08EA 1780
08EA 1781
08EA 1782
08EA 1783
08EA 1784
08EA 1785
08EA 1786
08EA 1787
08EA 1788
08EA 1789
08EA 1790
08EA 1791
08EA 1792
08EA 1793
08EA 1794
08EA 1795
08EA 1796
08EA 1797
08EA 1798
08EA 1799
08EA 1800
08EA 1801
08EA 1802
08EA 1803
08EA 1804
08EA 1805
08EA 1806
08EA 1807
08EA 1808
08EA 1809
08EA 1810
08EA 1811
08EA 1812
08EA 1813
08EA 1814
08EA 1815
08EA 1816
08EA 1817
08EA 1818
08EA 1819
08EA 1820
08EA 1821
08EA 1822
08EA 1823
08EA 1824
08EA 1825
08EA 1826
08EA 1827
08EA 1828
08EA 1829
08EA 1830
08EA 1831
08EA 1832
08EA 1833
08EA 1834
08EA 1835
08EA 1836
08EA 1837
08EA 1838
08EA 1839
08EA 1840
08EA 1841
08EA 1842
08EA 1843
08EA 1844
08EA 1845
08EA 1846
08EA 1847
08EA 1848
08EA 1849
08EA 1850
08EA 1851
08EA 1852
08EA 1853
08EA 1854
08EA 1855
08EA 1856
08EA 1857
08EA 1858
08EA 1859
08EA 1860
08EA 1861
08EA 1862
08EA 1863
08EA 1864
08EA 1865
08EA 1866
08EA 1867
08EA 1868
08EA 1869
08EA 1870
08EA 1871
08EA 1872
08EA 1873
08EA 1874
08EA 1875
08EA 1876
08EA 1877
08EA 1878
08EA 1879
08EA 1880
08EA 1881
08EA 1882
08EA 1883
08EA 1884
08EA 1885
08EA 1886
08EA 1887
08EA 1888
08EA 1889
08EA 1890
08EA 1891
08EA 1892
08EA 1893
08EA 1894
08EA 1895
08EA 1896
08EA 1897
08EA 1898
08EA 1899
08EA 1900
08EA 1901
08EA 1902
08EA 1903
08EA 1904
08EA 1905
08EA 1906
08EA 1907
08EA 1908
08EA 1909
08EA 1910
08EA 1911
08EA 1912
08EA 1913
08EA 1914
08EA 1915
08EA 1916
08EA 1917
08EA 1918
08EA 1919
08EA 1920
08EA 1921
08EA 1922
08EA 1923
08EA 1924
08EA 1925
08EA 1926
08EA 1927
08EA 1928
08EA 1929
08EA 1930
08EA 1931
08EA 1932
08EA 1933
08EA 1934
08EA 1935
08EA 1936
08EA 1937
08EA 1938
08EA 1939
08EA 1940
08EA 1941
08EA 1942
08EA 1943
08EA 1944
08EA 1945
08EA 1946
08EA 1947
08EA 1948
08EA 1949
08EA 1950
08EA 1951
08EA 1952
08EA 1953
08EA 1954
08EA 1955
08EA 1956
08EA 1957
08EA 1958
08EA 1959
08EA 1960
08EA 1961
08EA 1962
08EA 1963
08EA 1964
08EA 1965
08EA 1966
08EA 1967
08EA 1968
08EA 1969
08EA 1970
08EA 1971
08EA 1972
08EA 1973
08EA 1974
08EA 1975
08EA 1976
08EA 1977
08EA 1978
08EA 1979
08EA 1980
08EA 1981
08EA 1982
08EA 1983
08EA 1984
08EA 1985
08EA 1986
08EA 1987
08EA 1988
08EA 1989
08EA 1990
08EA 1991
08EA 1992
08EA 1993
08EA 1994
08EA 1995
08EA 1996
08EA 1997
08EA 1998
08EA 1999
08EA 2000
08EA 2001
08EA 2002
08EA 2003
08EA 2004
08EA 2005
08EA 2006
08EA 2007
08EA 2008
08EA 2009
08EA 2010
08EA 2011
08EA 2012
08EA 2013
08EA 2014
08EA 2015
08EA 2016
08EA 2017
08EA 2018
08EA 2019
08EA 2020
08EA 2021
08EA 2022
08EA 2023
08EA 2024
08EA 2025
08EA 2026
08EA 2027
08EA 2028
08EA 2029
08EA 2030
08EA 2031
08EA 2032
08EA 2033
08EA 2034
08EA 2035
08EA 2036
08EA 2037
08EA 2038
08EA 2039
08EA 2040
08EA 2041
08EA 2042
08EA 2043
08EA 2044
08EA 2045
08EA 2046
08EA 2047
08EA 2048
08EA 2049
08EA 2050
08EA 2051
08EA 2052
08EA 2053
08EA 2054
08EA 2055
08EA 2056
08EA 2057
08EA 2058
08EA 2059
08EA 2060
08EA 2061
08EA 2062
08EA 2063
08EA 2064
08EA 2065
08EA 2066
08EA 2067
08EA 2068
08EA 2069
08EA 2070
08EA 2071
08EA 2072
08EA 2073
08EA 2074
08EA 2075
08EA 2076
08EA 2077
08EA 2078
08EA 2079
08EA 2080
08EA 2081
08EA 2082
08EA 2083
08EA 2084
08EA 2085
08EA 2086
08EA 2087
08EA 2088
08EA 2089
08EA 2090
08EA 2091
08EA 2092
08EA 2093
08EA 2094
08EA 2095
08EA 2096
08EA 2097
08EA 2098
08EA 2099
08EA 2100
08EA 2101
08EA 2102
08EA 2103
08EA 2104
08EA 2105
08EA 2106
08EA 2107
08EA 2108
08EA 2109
08EA 2110
08EA 2111
08EA 2112
08EA 2113
08EA 2114
08EA 2115
08EA 2116
08EA 2117
08EA 2118
08EA 2119
08EA 2120
08EA 2121
08EA 2122
08EA 2123
08EA 2124
08EA 2125
08EA 2126
08EA 2127
08EA 2128
08EA 2129
08EA 2130
08EA 2131
08EA 2132
08EA 2133
08EA 2134
08EA 2135
08EA 2136
08EA 2137
08EA 2138
08EA 2139
08EA 2140
08EA 2141
08EA 2142
08EA 2143
08EA 2144
08EA 2145
08EA 2146
08EA 2147
08EA 2148
08EA 2149
08EA 2150
08EA 2151
08EA 2152
08EA 2153
08EA 2154
08EA 2155
08EA 2156
08EA 2157
08EA 2158
08EA 2159
08EA 2160
08EA 2161
08EA 2162
08EA 2163
08EA 2164
08EA 2165
08EA 2166
08EA 2167
08EA 2168
08EA 2169
08EA 2170
08EA 2171
08EA 2172
08EA 2173
08EA 2174
08EA 2175
08EA 2176
08EA 2177
08EA 2178
08EA 2179
08EA 2180
08EA 2181
08EA 2182
08EA 2183
08EA 2184
08EA 2185
08EA 2186
08EA 2187
08EA 2188
08EA 2189
08EA 2190
08EA 2191
08EA 2192
08EA 2193
08EA 2194
08EA 2195
08EA 2196
08EA 2197
08EA 2198
08EA 2199
08EA 2200
08EA 2201
08EA 2202
08EA 2203
08EA 2204
08EA 2205
08EA 2206
08EA 2207
08EA 2208
08EA 2209
08EA 2210
08EA 2211
08EA 2212
08EA 2213
08EA 2214
08EA 2215
08EA 2216
08EA 2217
08EA 2218
08EA 2219
08EA 2220
08EA 2221
08EA 2222
08EA 2223
08EA 2224
08EA 2225
08EA 2226
08EA 2227
08EA 2
```

```

097F 1040          .SBTTL MATCH PSECT OPTIONS FOR PRINTING
097F 1041
097F 1042 :++
097F 1043 : FUNCTIONAL DESCRIPTION:
097F 1044 :
097F 1045 :     THIS ROUTINE PRINTS EITHER 'XXX' OR 'NOXXX' FOR A GIVEN
097F 1046 :     PSECT OPTION.
097F 1047 :
097F 1048 : INPUTS:
097F 1049 :
097F 1050 :     R4     BIT VALUE
097F 1051 :     R5     OPTIONS
097F 1052 :
097F 1053 :--
097F 1054
097F 1055 PSECT_OPT MATCH:
53  50  54  B0 097F 1056      MOVW    R4,R0          ;COPY BITS
      08  13 0982 1057      BEQL    10$          ;IF EQL NOT COMPLEMENTED
      55  50  B3 0984 1058      BITW    R0,R5          ;NEGATIVE POLARITY?
      03  12 0987 1059      BNEQ    10$          ;IF NEQ NO
      50  50  B2 0989 1060      MCOMW   R0,R0          ;YES--GET COMPLEMENTED VALUE
      07 A3 50  B1 098C 1061 10$: MOVAB   L^PSC$G_OPTIONS,R3 ;POINT TO PSECT OPTIONS
      53  63  D0 0993 1062 20$: CMPW    R0,SYMS$_VAL+2(R3) ;IS THIS IT?
      F5  12 0997 1063      BEQL    30$          ;IF EQL YES
      50  D4 0999 1064      MOVL    SYMS$_LINK(R3),R3 ; No--link to next
      05  09 099C 1065      BNEQ    20$          ;IF NEQ GO ON
      04  09 099E 1066      CLRL    R0          ;RETURN 0 FOR NOT FOUND
      50  04 A3 09A0 1067      RSB
      50  53  50  9A 09A1 1068 30$: MOVZBL  SYMS$_NAME(R3),R0 ; Get offset to count/name
      51  80  C3 09A5 1069      SUBL3   R0,R3,R0      ; and form its address
      00000000'EF  D0 09AC 1071      MOVZBL  (R0)+,R1      ;COPY THE SIZE
      03  51  9A 09A9 1070      MOVZBL  L^MAC$GL_LIST_PTR,R2 ;GET LISTING POINTER
      30  BB 09B3 1072      PUSHR  #^M<R4,R5>    ;SAVE OPTION PARAMETERS
      02  14  91 09B5 1073      CMPB   R1,#3        ;IS IT LESS THAN 3-CHAR NAME?
      82  B5 09B8 1074      BGTR   40$          ;IF GTR NO
      62  60  51 28 09BA 1075      TSTW   (R2)+        ;YES--BUMP POINTER BY TWO
      00000000'EF  30 BA 09BC 1076 40$: MOVC3   R1,(R0),(R2) ;COPY INTO BUFFER
      50  01  C0 09C0 1077      POPR   #^M<R4,R5>    ;RESTORE OPTION PARAMS
      06  C0 09C2 1078      ADDL2  #6,L^MAC$GL_LIST_PTR ;TAB TO NEXT OPTION
      05  09  9A 09C9 1079      MOVZBL  #1,R0        ;RETURN FOUND
      05  09  05 09CC 1080      RSB

```

```

09CD 1082          .SBTTL PRINT SYMBOL INFORMATION FOR ONE SYMBOL
09CD 1083
09CD 1084          :++
09CD 1085          : FUNCTIONAL DESCRIPTION:
09CD 1086
09CD 1087          : THIS ROUTINE PRINTS THE SYMBOL INFORMATION FOR ONE SYMBOL
09CD 1088
09CD 1089          : INPUTS:
09CD 1090
09CD 1091          : R6 POINTS TO THE SYMBOL BLOCK
09CD 1092
09CD 1093          :--
09CD 1094
09CD 1095          MAC$PRT_SYM_INF:
09CD 1096          PUSHRR #^M<R6,R7,R8,R9,R10>
09D1 1097          MOVL L^MAC$GL_LIST_PTR,R10 ;GET CURRENT POINTER
09D8 1098          PUSHLL R10 ;COPY ONTO STACK
09DA 1099          MOVZBL SYMSB_NAME(R6),R0 ; Get offset to symbol count/name
09DE 1100          SUBL3 R0,R6,R0 ; and form its address
09E2 1101          MOVZBL (R0)+,R1 ; Get count and advance pointer to name
09E5 1102          MOVCS R1,(R0),(R10) ; Copy into listing buffer
09E9 1103          ADDL3 #<SYMSK_TWOCOL-1>,(SP),R10 ; Point past end of name
09ED 1104          BBC #FLG$V_SYM2COL,(R11),5$ ; Branch if not 2 column listing
09F1 1105          ADDL2 #<SYMSR_MAXLEN-SYMSK_TWOCOL+1>,R10 ; Allow for 31 character symbol
09F4 1106          5$:
09F4 1107          BBC #SYMSV_ASN,SYMSW_FLAG(R6),10$ ;BRANCH IF NOT ASSIGNED SYMBOL
09F9 1108          MOVB #^A/=/, (R10)+ ;YES--STORE EQUAL SIGN
09FC 1109          BRB 20$
09FE 1110          10$:
0A01 1111          20$:
0A01 1111          ADDL2 #9,R10 ;MAKE ROOM FOR VALUE
0A04 1112          MOVL R10,L^MAC$GL_LIST_PTR ;STORE CURRENT POINTER
0A0B 1113          PUSHLL R10 ;SAVE ON STACK ALSO
0A0D 1114          BBS #SYMSV_DEF,SYMSW_FLAG(R6),30$ ;BRANCH IF SYMBOL DEFINED
0A12 1115          MOVCS #0,(SP),#^A/*/,#B,-8(R10) ;NO--FILL VALUE FIELD WITH STARS
0A19 1116          BRB 40$
0A1B 1117          30$:
0A1D 1118          INCL R7 ;DEFINED--STACK VALUE
0A26 1119          MOVL SYMSL_VAL(R6),L^MAC$AL_VALSTACK[R7] ;...
0A29 1120          MOVZBL #4,R0 ;LIST LONGWORD VALUE
0A2C 1121          40$:
0A2F 1122          BSBW MAC$LIST_BYT_0 ;...
0A33 1123          POPL R1C ;RESTORE POINTER
0A37 1124          MOVZWL SYMSW_FLAG(R6),R0 ;GET SYMBOL FLAGS
0A3B 1125          50$:
0A3F 1126          BBC #SYMSV_WEAK,R0,50$ ;BRANCH IF NOT WEAK
0A43 1127          MOVB #^A/W/,0(R10) ;YES--FLAG IT
0A43 1128          BBS #SYMSV_ABS,R0,60$ ;BRANCH IF ABSOLUTE SYMBOL
0A43 1129          BBC #SYMSV_DEF,R0,60$ ;NO--BRANCH IF DEFINED
0A43 1129          ;
0A43 1128          ; SYMBOL IS RELOCATABLE AND DEFINED
0A43 1129          ;
0A43 1130          60$:
0A48 1131          MOVB #^A/R/,1(R10) ;FLAG RELOCATABLE
0A4C 1132          BBC #SYMSV_GLOBL,R0,70$ ;BRANCH IF NOT GLOBAL SYMBOL
0A51 1133          70$:
0A55 1134          MOVB #^A/G/,2(R10) ;YES--FLAG IT
0A5A 1135          BBC #SYMSV_EXTRN,R0,80$ ;BRANCH IF NOT EXTERNAL
0A5E 1136          80$:
0A63 1137          MOVB #^A/X/,3(R10) ;YES--FLAG IT
0A66 1138          BBC #SYMSV_ODBG,R0,90$ ;BRANCH IF NO DEBUG OUTPUT FOR SYMBOL
0A66 1138          BITW #^A/D/,4(R10) ;YES--FLAG IT
0A66 1138          90$:
0A66 1138          BNEQ #SYMSM_DEF!SYMSM_GLOBL!SYMSM_EXTRN,R0 ;
0A66 1138          100$ ;IF DEFINED, GLOBAL OR EXTERNAL

```

```

05 AA 55 8F 90 0A68 1139      MOVB   #^A/U/,5(R10)      ;NO--MARK AS UNDEFINED
          5A 09  C0 0A6D 1140 100$: ADDL2  #9,R10           ;MOVE TO PSECT SPOT
00000000'EF 5A  D0 0A70 1141      MOVL   R10,L^MAC$GL_LIST_PTR ;STORE POINTER
          20 50  E0 0A77 1142      BBS    #SYMSV_ABS,R0,110$   ;BRANCH IF ABSOLUTE
          50 0C A6 9A 0A7B 1143      MOVZBL SYMSB_SEG(R6),R0    ;GET SEGMENT NUMBER
          0A 13 07  E0 0A81 1145      BBS    #SYMSV_REF,-        ;DON'T FIDDLE WITH THE ABS PSECT.
02 00000009'GF 50  D7 0A89 1147      DECL  R0                   ;HAS THE BLANK PSECT BEEN FILTERED OUT?
          57  D6 0A8B 1148 105$: INCL  R7                   ;YES, ADJUST THE PSECT# IN SYMBOL TABLE.
00000000'EF47 50  D0 0A8D 1149      MOVL   R0,L^MAC$AL_VALSTACK[R7] ;STACK IT
          50 01 9A 0A95 1150      MOVZBL #1,R0                ;LIST ONE BYTE
          F565' 30 0A98 1151      BSBW   MAC$LIST_BYT 0      ;
00000000'EF 8E 28  C1 0A9B 1152 110$: ADDL3  #40,(SP),L^MAC$GL_LIST_PTR ; Figure new pointer and store it
          07 6B 2A  E1 0AA3 1153      BBC    #FLGSV_SYM2COL,(R1T),120$ ; Branch if not 2 column listing
00000000'EF 12  C0 0AA7 1154      ADDL2  #18,L^MAC$GL_LIST_PTR ; Allow for larger symbol name
          07C0 8F BA 0AAE 1155 120$: POPR  #^M<R6,R7,R8,R9,R10> ;RESTORE REGS
          05 05 0AB2 1157      RSB
          0AB3 1158
          0AB3 1159 MAC$LIST_INIT:
50 00000000'EF 9E 0AB3 1160      MOVAB  L^MAC$AB_LST_END,R0   ;POINT TO BUFFER
00000000'EF 50  D0 0ABA 1161      MOVL   R0,L^MAC$GL_LIST_PTR ;SET POINTER
60 0084 8F 20 6E 00 2C 0AC1 1162      MOVCS  #0,(SP),#^A7 /, #T32,(R0);FILL BUFFER WITH BLANKS
          05 0AC9 1163      RSB
          OACA 1164
          OACA 1165      .END

```

AUD\$K_SIZE = 00000010
BLNK = 00000020
CHRSM_COMMA CR = 00000020
CHRSM_ILL_CR = 00000040
CHRSM_NUM_BER = 00000010
CHRSM_SPA_MSK = 00000001
CHRSM_SYM_CH1 = 00000008
CHRSM_SYM_CHR = 00000004
CHRSM_SYM_DLM = 00000002
CHR\$V_COMMA CR = 00000005
CHR\$V_CVTLWC = 00000061
CHR\$V_ILL_CHR = 00000006
CHR\$V_NOCVT = 0000007F
CHR\$V_NUM_BER = 00000004
CHR\$V_SPA_MSK = 00000000
CHR\$V_SYM_CH1 = 00000003
CHR\$V_SYM_CHR = 00000002
CHR\$V_SYM_DLM = 00000001
CR = 0000000D
CREF_TREE_OUT = 0000078F R 04
CRFSK_DELETE = ***** X 04
CRFSL_K1FMTBL = 00000048
CRFSM_DEFAULT = 00000012
CRFSM_DIR = 00000001
CRFSM_MACROS = 00000002
CRFSM_OPCODES = 00000004
CRFSM_REGISTERS = 00000008
CRFSM_SYMBOLS = 00000010
CRFSOOT = ***** X 04
CRFSV_DIR = 00000000
CRFSV_MACROS = 00000001
CRFSV_OPCODES = 00000002
CRFSV_REGISTERS = 00000003
CRFSV_SYMBOLS = 00000004
DBGSC_LIT_DAT = 00000000
DBGSC_MEND = 000000BD
DBGSC_MODULE = 000000BC
DBGSC_PSECT = 00000088
DBGSC_REL_DAT = 00000001
DBGSC_ROUTINE = 000000BE
DBGSC_SYMBOL = 000000BA
DBGSK_MEND_LEN = 00000002
DBGSK_MODULE_LN = 00000008
DBGSK_PSECT_LEN = 0000000C
DBGSK_ROUTIN_LN = 00000008
DBGSK_SYMBOL_LN = 00000008
ENBSG_GLOBAL = ***** X 04
ENBSG_TRACEBACK = ***** X 04
EOMSC_ABORT = 00000003
EOMSC_ERROR = 00000002
EOMSC_SUCCESS = 00000000
EOMSC_WARNING = 00000001
FF = 0000000C
FIN_ASM_EXIT = 00000787 R 04
FLGSM_ALLCHR = 00000001
FLGSM_BOL = 00000002
FLGSM_CHKLPND = 00100000

FLGSM_COMPEXPR = 00000004
FLGSM_CONT = 00000008
FLGSM_CRF = 40000000
FLGSM_CRSEEN = 00000001
FLGSM_DATRPT = 00000010
FLGSM_DBGOUT = 00004000
FLGSM_DLIMSTR = 00008000
FLGSM_ENDMCH = 00000020
FLGSM_EVALEXPR = 00000040
FLGSM_EXPOPT = 00000080
FLGSM_EXTERR = 00010000
FLGSM_EXTWRN = 00020000
FLGSM_FIRSTLN = 00000200
FLGSM_IFSTAT = 00800000
FLGSM_IIF = 00400000
FLGSM_INSERT = 00000100
FLGSM_IRPC = 20000000
FLGSM_LEXOP = 00000002
FLGSM_LSTXST = 00000200
FLGSM_MAC2COL = 00000800
FLGSM_MACL = 00000800
FLGSM_MACLTB = 08000000
FLGSM_MACTXT = 00010000
FLGSM_MEBLST = 00001000
FLGSM_MOREARG = 00002000
FLGSM_MOREINP = 00000008
FLGSM_NEWPND = 00000400
FLGSM_NOREF = 01000000
FLGSM_NTTYPEPC = 00000020
FLGSM_NULCHR = 00040000
FLGSM_OBJXST = 00200000
FLGSM_OPNDCHK = 00000100
FLGSM_OPRND = 00002000
FLGSM_OPTVFLIDX = 00001000
FLGSM_ORDLST = 00020000
FLGSM_P2 = 00004000
FLGSM_RPTIRP = 10000000
FLGSM_SEQFIL = 02000000
FLGSM_SKAN = 00008000
FLGSM_SPECOP = 00000004
FLGSM_SPLALL = 04000000
FLGSM_STOIMF = 00040000
FLGSM_SYM2COL = 00000400
FLGSM_TOCFLG = 00080000
FLGSM_UPAFLG = 00000010
FLGSM_UPDFIL = 00000080
FLGSM_UPMARG = 00000040
FLGSM_XCRF = 80000000
FLG\$V_ALLCHR = 00000000
FLG\$V_BOL = 00000001
FLG\$V_CHKLPND = 00000014
FLG\$V_COMPEXPR = 00000002
FLG\$V_CONT = 00000003
FLG\$V_CRF = 0000001E
FLG\$V_CRSEEN = 00000020
FLG\$V_DATRPT = 00000004
FLG\$V_DBGOUT = 0000002E

MAC\$FINISH
Symbol table

ROUTINES FOR FINISHING ASSEMBLY

E 10

16-SEP-1984 02:15:54 VAX/VMS Macro V04-00
5-SEP-1984 01:48:10 [MACRO.SRC]FINISH.MAR;1

Page 29
(19)

MA
VO

FLGSV_DLIMSTR	=	0000002F	MAC\$AB_IFP_FAO	*****	X	04
FLGSV_ENDMCH	=	00000005	MAC\$AB_IND_HDR	*****	X	04
FLGSV_EVALEXPR	=	00000006	MAC\$AB_INJ_HDRA	*****	X	04
FLGSV_EXPOPT	=	00000007	MAC\$AB_IND_HDRB	*****	X	04
FLGSV_EXTERR	=	00000030	MAC\$AB_LINEBF	*****	X	04
FLGSV_EXTWRN	=	00000031	MAC\$AB_LINE_END	*****	X	03
FLGSV_FIRSTLN	=	00000029	MAC\$AB_LPG_FMT	*****	X	04
FLGSV_IFSTAT	=	00000017	MAC\$AB_LST_END	*****	X	03
FLGSV_IIF	=	00000016	MAC\$AB_MCD_FAO	*****	X	04
FLGSV_INSERT	=	00000008	MAC\$AB_MLB_FAO	*****	X	04
FLGSV_IRPC	=	0000001D	MAC\$AB_MLB_HD2	*****	X	04
FLGSV_LEXOP	=	00000021	MAC\$AB_MLB_HD2A	*****	X	04
FLGSV_LSTXST	=	00000009	MAC\$AB_MLB_HDR	*****	X	04
FLGSV_MAC2COL	=	0000002B	MAC\$AB_MLB_HDRA	*****	X	04
FLGSV_MACL	=	0000000B	MAC\$AB_MLB_HDRB	*****	X	04
FLGSV_MACLTB	=	0000001B	MAC\$AB_MLB_SUM	*****	X	04
FLGSV_MACTXT	=	00000010	MAC\$AB_MLB_TOT	*****	X	04
FLGSV_MEBLST	=	0000000C	MAC\$AB_NOERRM	*****	X	04
FLGSV_MOREARG	=	0000002D	MAC\$AB_OBSR_FAO	*****	X	04
FLGSV_MOREINP	=	00000023	MAC\$AB_PSC_RD2	*****	X	04
FLGSV_NEWPND	=	0000000A	MAC\$AB_PSC_HD2A	*****	X	04
FLGSV_NOREF	=	00000018	MAC\$AB_PSC_HDR	*****	X	04
FLGSV_NTTYPEPC	=	00000025	MAC\$AB_PSC_HDRA	*****	X	04
FLGSV_NULCHR	=	00000032	MAC\$AB_PSC_HDRB	*****	X	04
FLGSV_OBJXST	=	00000015	MAC\$AB_PSS_FAO	*****	X	04
FLGSV_OPNDCHK	=	00000028	MAC\$AB_PSS_MSG	*****	X	04
FLGSV_GPRND	=	0000000D	MAC\$AB_RNT_MSG	*****	X	04
FLGSV_OPTVFLIDX	=	0000002C	MAC\$AB_RUN_HDR	*****	X	04
FLGSV_ORDLST	=	00000011	MAC\$AB_RUN_HDRA	*****	X	04
FLGSV_P2	=	0000000E	MAC\$AB_SBT_IDNT	*****	X	04
FLGSV_RPTIRP	=	0000001C	MAC\$AB_STB_MSG	*****	X	04
FLGSV_SEQFIL	=	00000019	MAC\$AB_SYP_FAO	*****	X	04
FLGSV_SKAN	=	0000000F	MAC\$AB_TIT[E	*****	X	04
FLGSV_SPECOP	=	00000022	MAC\$AB_WAS_MSG	*****	X	04
FLGSV_SPLALL	=	0000001A	MAC\$AB_WERE_MG	*****	X	04
FLGSV_STOIMF	=	00000012	MAC\$AB_WSL_FAO	*****	X	04
FLGSV_SYM2COL	=	0000002A	MAC\$AL_CRF_TB1	*****	X	04
FLGSV_TOCFLG	=	00000013	MAC\$AL_CRF_TB1A	*****	X	04
FLGSV_UPAFLG	=	00000024	MAC\$AL_CRF_TB1B	*****	X	04
FLGSV_UPDFIL	=	00000027	MAC\$AL_CRF_TB2	*****	X	04
FLGSV_UPMARG	=	00000026	MAC\$AL_CRF_TB2A	*****	X	04
FLGSV_XCRF	=	0000001F	MAC\$AL_CRF_TB2W	*****	X	04
HASHSZ	=	0000007F	MAC\$AL_CRF_TB2WA	*****	X	04
HYPHEN	=	0000002D	MAC\$AL_CRF_TB3	*****	X	04
INPSK_BUFSIZ	=	000003E8	MAC\$AL_CRF_TB5	*****	X	04
INTSK_BUFSIZ	=	000013F4	MAC\$AL_CRF_TB7	*****	X	04
INTSK_BUFWRN	=	00001390	MAC\$AL_CRF_TB8	*****	X	04
LSTSK_BUFSIZ	=	00000086	MAC\$AL_CRF_TB9	*****	X	04
LSTSK_L_P_PAGE	=	0000003C	MAC\$AL_VALSTACK	*****	X	04
LSTSK_TIT[E_SIZ	=	00000028	MAC\$AW_IND_NAMS	*****	X	04
MAC\$AB_CMD_FAO	*****	X	MAC\$AW_RNT_PTRS	*****	X	04
MAC\$AB_CRF_MSG	*****	X	MAC\$AW_TIM_PTRS	*****	X	04
MAC\$AB_CRF_TB4	*****	X	MAC\$CHKBYT	*****	X	04
MAC\$AB_CRF_TB6	*****	X	MAC\$DBG_NAM_OUT	00000854	R	04
MAC\$AB_DEF_TITL	*****	X	MAC\$DBG_PSECT	00000892	R	04
MAC\$AB_ERRM	*****	X	MAC\$DBG_VAL_OUT	00000866	R	04
MAC\$AB_FAO_TIM	*****	X	MAC\$DEC_OUT_L2R	*****	X	04

! 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
\$AB\$\$	00000177 (375.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
MAC\$RW_DATA	00000008 (8.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
MAC\$RO_CODE_P3	00000ACA (2762.)	04 (4.)	NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:00.96
Command processing	106	00:00:00.35	00:00:03.59
Pass 1	428	00:00:10.26	00:00:43.38
Symbol table sort	0	00:00:01.28	00:00:05.53
Pass 2	237	00:00:02.65	00:00:13.49
Symbol table output	51	00:00:00.25	00:00:01.32
Psect synopsis output	2	00:00:00.02	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	855	00:00:14.88	00:01:08.29

The working set limit was 1800 pages.
88811 bytes (174 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1272 non-local and 115 local symbols.
1165 source lines were read in Pass 1, producing 36 object records in Pass 2.
43 pages of virtual memory were used to define 42 macros.

! Macro library statistics !

Macro library name	Macros defined
-\$25;\$DUA28:[MACRO.OBJ]MACRO.MLB;1	8
-\$25;\$DUA28:[SYSLIB]STARLET.MLB;2	10
TOTALS (all libraries)	18

1364 GETS were required to define 18 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:FINISH/OBJ=OBJ\$:FINISH MSRC\$:FINISH/UPDATE=(ENH\$:FINISH)+LIB\$:MACRO/LIB

The image displays a grid of 100 small terminal window screenshots, arranged in a 10x10 grid. Each window shows a different VAX/VMS command and its output. The windows are arranged in a grid, with some windows clearly legible and labeled with their command names. The labels are as follows:

- DATA LIS (top-left)
- DEFINE LIS (middle-left)
- FLOAT LIS (middle-right)
- ERRMSG LIS (lower-middle)
- GETARG LIS (lower-right)
- DATA LIS (bottom-left)
- INPUT LIS (bottom-right)
- ERROR LIS (bottom-center)
- FINISH LIS (bottom-center)
- GETCMD LIS (bottom-center)

The other windows in the grid show various system outputs, including file listings, directory structures, and command execution results. The text is small and dense, typical of a terminal window output.