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
```

```
**FILE**ID**GETCMD                                                  N 12                                                   MAC
                                                                                                                           V04

  GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT    CCCCCCCC  MM      MM  DDDDDDDD
  GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT    CCCCCCCC  MM      MM  DDDDDDDD
GG          EE              TT      CC          MMMM  MMMM  DD      DD
GG          EE              TT      CC          MMMM  MMMM  DD      DD
GG          EE              TT      CC          MM  MM  MM  DD      DD
GG          EE              TT      CC          MM  MM  MM  DD      DD
GG          EEEEEEEE        TT      CC          MM      MM  DD      DD
GG          EEEEEEEE        TT      CC          MM      MM  DD      DD
GG  GGGGGG  EE              TT      CC          MM      MM  DD      DD
GG  GGGGGG  EE              TT      CC          MM      MM  DD      DD
GG      GG  EE              TT      CC          MM      MM  DD      DD      ....
GG      GG  EE              TT      CC          MM      MM  DD      DD      ....
  GGGGGG    EEEEEEEEEE      TT        CCCCCCCC  MM      MM  DDDDDDDD        ....
  GGGGGG    EEEEEEEEEE      TT        CCCCCCCC  MM      MM  DDDDDDDD        ....

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
```

Table of contents

```
0000           .TITLE  MAC$GETCMD      GET COMMAND 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: 28-AUG-78
0000   ;
0000   ; MODIFIED BY:
0000   ;
0000   ;       V03-010         RRB0031         Rowland R. Bradley      15-Jul-1984
0000   ;               Reinit dynamic string desc. prior to calling LIST_QUAL.  This
0000   ;               avoids trashing the output filename.
0000   ;
0000   ;       V03-009         RRB0030         Rowland R. Bradley      06-Jul-1984
0000   ;               Fix concatenated file lists.  Handle library files in
0000   ;               searchlists(use current global context) and flag errors
0000   ;               appropriately.  Enable user to specify object and listing
0000   ;               in different places.  Replace NAM block(it is needed by
0000   ;               SUM$xxx routines).  Fix problem with /UPDATES to properly
0000   ;               handle an update file.
0000   ;
0000   ;       V03-008         RRB0025         Rowland R. Bradley      21-May-1984
0000   ;               Add support for proper handling of input files
0000   ;               using LIB$FIND_FILE.  Add appropriate comments and
```

```
0000  ;                 clean up existing comments.  Signal LIB$FIND_FILE errors
0000  ;                 correctly.  Reorganize GET_INPUT to more efficiently handle
0000  ;                 input files.  Handle positional qualifiers on libraries.
0000  ;
0000  ;         V03-007         RRB0014         Rowland R. Bradley      27-Feb-1984
0000  ;                 Bug fix which handles positional /LIST qualifiers on libraries
0000  ;                 as a global /LIST qualifier.
0000  ;
0000  ;         V03-006         RRB0011         Rowland R. Bradley      29-Jan-1984
0000  ;                 Modify to handle locally present /LIST & /OBJECT qualifiers
0000  ;                 correctly(added GET_LOC_LIST, GET_LOC_OBJECT, and modified
0000  ;                 GET_LIST and GET_OBJECT).  Fix length of filename for
0000  ;                 to handle long filenames in INP_FILE.
0000  ;
0000  ;         V03-005         MTR0030         Mike Rhodes             28-Mar-1983
0000  ;                 Modify code to handle GLOBAL vs. LOCALly PRESENT qualifiers.
0000  ;
0000  ;         V03-004         MTR0029         Mike Rhodes             18-Mar-1983
0000  ;                 Check for explicit negation of qualifiers, before performing
0000  ;                 the negation actions (due to new CLI interface).
0000  ;
0000  ;         V03-003         MTR0026         Mike Rhodes             24-Feb-1983
0000  ;                 Remember the length of the last input file opened with its
0000  ;                 expanded string in MAC$INP_NAM_BUF.  The cell holding the
0000  ;                 length is MAC$GB_INPNAMLEN.
0000  ;
0000  ;         V03-002         MTR0022         Mike Rhodes             01-Feb-1983
0000  ;                 Convert to new CLI interface.
0000  ;
0000  ;         V03-001         MTR0020         Mike Rhodes             07-Jul-1982
0000  ;                 Modify the handling of the /ENABLE /DISABLE /SHOW and /NOSHOW
0000  ;                 qualifiers, such that they truly override their corresponding
0000  ;                 assembler directives actions.  A new flag MAC$GL_DSLISF has
0000  ;                 been employed to facilitate this manipulation.
0000  ;                 Affected modules are SHOW_ENAB and DBG_NEXT_OPT.
0000  ;
0000  ;         V03-000         MTR0010         Mike Rhodes             15-Mar-1982
0000  ;                 Remove $CLIDEFQUALMACR macro, since the symbols which it
0000  ;                 defined are globally available during the build.
0000  ;
0000  ;         V02-014         MTR0002         Mike Rhodes             01-Dec-1981
0000  ;                 Modify the /DEBUG action routine to allow the SYMBOLS and
0000  ;                 TRACEBACK qualifier values to coexist in a qualifier list
0000  ;                 without negating one another.
0000  ;
0000  ;         V02-013         MTR0001         Mike Rhodes             09-Nov-1981
0000  ;                 Add a new action routine (called by the CLI parsing routines)
0000  ;                 to process the /[NO]DEBUG qualifier and its parameters.  The
0000  ;                 processing consists of setting/clearing bits in the options
0000  ;                 vector MAC$GL_ENLISF.
0000  ;
0000  ;         V02-012         BLS0057         Benn Schreiber          13-Jun-1981
0000  ;                 Correct reference to SUM$OPEN to use General addressing mode
0000  ;
0000  ;--
```

```
                                         0000   114         .SBTTL  DECLARATIONS
                                         0000   115 ;
                                         0000   116 ; INCLUDE FILES:
                                         0000   117 ;
                                         0000   118
                                         0000   119 ;
                                         0000   120 ; MACROS:
                                         0000   121 ;
                                         0000   122
                                         0000   123         $MAC_MLFDEF                     ; Define MLF offsets
                                         0177   124         $MAC_CTLFLGDEF                  ;DEFINE VAX MACRO CONTROL FLAGS
                                         0177   125         $MAC_SYMBLKDEF                  ;DEFINE MAX LENGTH OF SYMBOL NAME
                                         0000   126         $MAC_CRFLAGDEF                  ;DEFINE CREF FLAG BITS
                                         0000   127         $CLIDEF                         ;CLI DEFINITIONS
                                         0000   128         $DSCDEF                         ;DESCRIPTOR DEFINITIONS
                                         0000   129         $FABDEF                         ;DEFINE FAB OFFSETS
                                         0000   130         $NAMDEF                         ;DEFINE NAM BLOCK OFFSETS
                                         0000   131
                                         0000   132 ;
                                         0000   133 ; EQUATED SYMBOLS:
                                         0000   134 ;
                                         0000   135
                                         0000   136         _VIELD  MAC,1,<-
                                         0000   137                 LIST,-                  ;LIST FILE PRESENT FLAG
                                         0000   138                 OBJECT,-                ;OBJECT FILE PRESENT FLAG
                                         0000   139                 INPFIL,-                ;INPUT FILE IS PRESENT
                                         0000   140                 CROSSREF,-              ;CROSS REFERENCE REQUESTED
                                         0000   141                 <LIBRARY,,M>, -         ;Input file is a library
                                         0000   142                 <UPDATE,,M>, -          ;Input file has updates
                                         0000   143                 >
                                         0000   144
                                         0000   145 ;
                                         0000   146 ; OWN STORAGE:
                                         0000   147 ;
                                     00000000   148         .PSECT  MAC$RW_DATA,NOEXE,RD,WRT
                                         0000   149
                            00000004     0000   150 UPDATE_LIST:    .BLKL   1                       ; Update list address.
                            00000000     0004   151 LOC_LST_FLAG:   .LONG   0                       ; Positional Flags to
                            00000000     0008   152 LOC_OBJ_FLAG:   .LONG   0                       ;   used to save the
                                         000C   153                                                 ;   positional /NOLIST,
                                         000C   154                                                 ;   /LIST, /NOOBJ, or /OBJ
                                         000C   155                                                 ;  status.
                                         000C   156
                                     00000000   157         .PSECT  MAC$RO_DATA,NOEXE,NOWRT,GBL,LONG
                                         0000   158
                            00000000     0000   159 RESULT:         .LONG                           ;ADDRESS OF RESULT DESC
                                         0004   160                                                 ;
                     52 41 4D 2E 00'     0004   161 INP_DEFEXT:     .ASCIC  /.MAR/                  ;DEFAULT FILE SPEC FOR INPUT
                                  04     0004
                                         0009   162                                                 ;
                                         0009   163 ;
                                         0009   164 ; CLI PARSING INFORMATION
                                         0009   165 ;
   45 4E 49 4C 24 00000011'010E0000'     0009   166 COMMAND_LINE:   .ASCID  /$LINE/
5F 53 53 4F 52 43 0000001E'010E0000'     0016   167 CROSS:          .ASCID  /CROSS_REFERENCE/
          45 43 4E 45 52 45 46 45 52     0024
   47 55 42 45 44 00000035'010E0000'     002D   168 DEBUG:          .ASCID  /DEBUG/
```

```
4C 42 41 53 49 44 00000042'010E0000' 003A    169 DISABLE:          .ASCID  /DISABLE/
                              45  0048
45 4C 42 41 4E 45 00000051'010E0000' 0049    170 ENABLE:           .ASCID  /ENABLE/
      45 4C 49 46 0000005F'010E0000' 0057    171 FILE:             .ASCID  /FILE/
52 41 52 42 49 4C 0000006B'010E0000' 0063    172 LIBRARY:          .ASCID  /LIBRARY/
                              59  0071
4E 49 54 53 49 4C 0000007A'010E0000' 0072    173 LISTING:          .ASCID  /LISTING/
                              47  0080
57 4F 48 53 4F 4E 00000089'010E0000' 0081    174 NOSHOW:           .ASCID  /NOSHOW/
54 43 45 4A 42 4F 00000097'010E0000' 008F    175 OBJECT:           .ASCID  /OBJECT/
      57 4F 48 53 000000A5'010E0000' 009D    176 SHOW:             .ASCID  /SHOW/
45 54 41 44 50 55 000000B1'010E0000' 00A9    177 UPDATE:           .ASCID  /UPDATE/
                                  00B7    178
                                  00B7         .SBTTL  PARSE COMMAND LINE
                                  00B7   ;++
                                  00B7   ; Functional Description:
                                  00B7   ;
                                  00B7   ;     This routine obtains the command line and the associated input,
                                  00B7   ;     object, and listing file(s) information supplied by call backs
                                  00B7   ;     to the CLI.
                                  00B7   ;
                                  00B7   ; Implicit Inputs:
                                  00B7   ;
                                  00B7   ;     Global Data (et.al.)
                                  00B7   ;
                                  00B7   ; Implicit Outputs:
                                  00B7   ;
                                  00B7   ;     The input file(s) are validated and set up in the input queue
                                  00B7   ;     MAC$GL_INPQUE, the library file(s) have been initialized, all
                                  00B7   ;     of the qualifier(s) have been parsed, the update file(s) have
                                  00B7   ;     been set up, and the output and listing file(s) have been created
                                  00B7   ;     (if they were requested).
                                  00B7   ;--
                              00000000         .PSECT  MAC$RO_CODE_P35,NOWRT,GBL,LONG
                                  0000
                                  0000   MAC$GETCMD::
            0F80 8F       BB      0000         PUSHR   #^M<R7,R8,R9,R10,R11>   ; SAVE REGISTERS
      00000000'EF         9F      0004         PUSHAB  L^MAC$GQ_RNT_CMD        ; STACK TIMING BLOCK ADDRESS
  00000000'GF      01     FB      000A         CALLS   #1,G^MAC$TIMER_ON       ; BEGIN TIMING COMMAND PROCESSING
                                  0011   ;
                                  0011   ; Create a dynamic string descriptor for parsing the command line.
                                  0011   ;
       5E    F8 AE        9E      0011         MOVAB   -8(SP), SP              ; Create a dynamic string descriptor
                   6E     7C      0015         CLRQ    (SP)                    ; which will be used for parsing the
                   02     90      0017         MOVB    #DSC$K_CLASS_D,-        ; command line.  It will be passed to
                03 AE             0019                 DSC$B_CLASS (SP)        ; the various processing routines.
          23 6B    23     E4      001B         BBSC    #FLG$V_MOREINP,(R11),10$; BR IF THIS IS NOT THE FIRST CALL
                                  001F
                                  001F   ;
                                  001F   ; Obtain a local copy of the command line.
                                  001F   ;
                   5E     DD      001F         PUSHL   SP                      ; Address of dynamic string descriptor.
      00000009'EF         9F      0021         PUSHAB  COMMAND_LINE            ; '$LINE' CLI keyword to get command line.
  00000000'GF      02     FB      0027         CALLS   #2, G^CLI$GET_VALUE     ; Get the command line locally.
  00000000'GF      6E     3C      002E         MOVZWL  (SP), G^MAC$GL_CMDLEN   ; Save the length of the command line
                04 AE     D0      0035         MOVL    DSC$A_POINTER (SP),-    ; and its address for later printing
      00000000'GF                 0038                 G^MAC$GL_CMDLIN         ; on the LISTING and/or the TERMINAL.
                04 AE     D4      003D         CLRL    DSC$A_POINTER (SP)      ; Reset address for next segment.
                   6E     B4      0040         CLRW    DSC$W_LENGTH (SP)       ; Reset length for next segment.
                                  0042
                                  0042   ;
                                  0042   ; Process the INPUT file(s) specification(s)     (plus:  /CROSS_REFERENCE     )
                                  0042   ;                                                (       /DEBUG   /DISABLE    )
                   5E     DD      0042   10$:  PUSHL   SP                      ;        (       /ENABLE  /LIBRARY    )
  0000035E'EF      01     FB      0044         CALLS   #1, GET_INPUT           ;        (       /LISTING /OBJECT     )
                                  004B                                         ;        (       /SHOW    /NOSHOW     )
                                  004B                                         ;        (       /UPDATE              )
                                  004B   ;
                                  004B   ; TERMINATE PARSE
                                  004B   ;
```

```
50    00000000'GF   9E  004B          MOVAB   G^MAC$GL_INPQUE,R0          ; POINT TO THE INPUT QUEUE
      50   04 A0    D1  0052          CMPL    4(R0),R0                    ; WERE THERE ANY INPUT FILES?
              11    12  0056          BNEQ    OPEN_OUTPUTS                ; IF NEQ YES--GO AHEAD
50    00000000'GF   9E  0058          MOVAB   G^MAC$MSG_NO_FILE,R0        ; NO--GET MESSAGE ADDRESS
00000000'GF   00    FB  005F          CALLS   #0,G^MAC$ERR_TEXT           ; REPORT THE ERROR
           FF97'    31  0066          BRW     MAC$LAST_CHANCE             ; AND GO DIE
                        0069
                        0069    ;
                        0069    ; OPEN OBJECT AND LISTING FILES IF REQUESTED
                        0069    ;
                        0069    OPEN_OUTPUTS:
5A    00000000'GF   9E  0069          MOVAB   G^MAC$GL_CTLMSK,R10         ; POINT TO CONTROL MASK
     34 6A    02    E1  0070          BBC     #MAC_V_OBJECT,(R10),20$ ; BRANCH IF NO OBJECT FILE
                        0074
                        0074          $CREATE FAB=G^MAC$OBJECT_FAB,-  ; CREATE THE OBJECT FILE
                        0074                  ERR=G^MAC$ERR_OPN_OUT
           73 50    E9  0087          BLBC    R0,40$                      ; BRANCH IF OPEN ERROR
                        008A          $CONNECT RAB=G^MAC$OBJECT_RAB,- ; CONNECT THE RECORD STREAM
                        008A                  EPR=G^MAC$ERR_OPN_OUT
           5D 50    E9  009D          BLBC    R0,40$                      ; BRANCH IF CONNECT ERROR
              15    E3  00A0          BBCS    #FLG$V_OBJXST,-             ; FLAG OBJECT FILE OPEN
  00 00000000'GF        00A2                  G^MAC$GL_FLAGS, 20$
                        00A8
     34 6A    01    E1  00A8    20$:  BBC     #MAC_V_LIST,(R10),30$       ; BRANCH IF NO LISTING FILE
                        00AC          $CREATE FAB=G^MAC$LIST_FAB,-    ; CREATE THE LISTING FILE
                        00AC                  ERR=G^MAC$ERR_OPN_OUT
           3B 50    E9  00BF          BLBC    R0,40$                      ; BRANCH IF OPEN ERROR
                        00C2          $CONNECT RAB=G^MAC$LIST_RAB,-   ; CONNECT THE RECORD STREAM
                        00C2                  ERR=G^MAC$ERR_OPN_OUT
           25 50    E9  00D5          BLBC    R0,40$                      ; BRANCH IF CONNECT ERROR
              09    E3  00D8          BBCS    #FLG$V_LSTXST,-             ; FLAG THERE IS A LISTING FILE
  00 00000000'GF        00DA                  G^MAC$GL_FLAGS, 30$
                        00E0
     04 6A    04    E1  00E0    30$:  BBC     #MAC_V_CROSSREF,(R10),35$ ; BRANCH IF NOT CROSS-REFERENCE
     00 6B    1E    E2  00E4          BBSS    #FLG$V_CRF,(R11),35$        ; YES--SET IN FLAGS WORD
                        00E8
     00000000'EF    9F  00E8    35$:  PUSHAB  L^MAC$GQ_RNT_CMD            ; STACK TIMING BLOCK ADDRESS
00000000'GF   01    FB  00EE          CALLS   #1,G^MAC$TIMER_OFF          ; STOP TIMING COMMAND PROCESSING
                        00F5
         5E   08    C0  00F5          ADDL    #8, SP                      ; Clean up the stack.
       0F80 8F      BA  00F8          POPR    #^M<R7,R8,R9,R10,R11>       ; RESTORE REGISTERS
                    05  00FC          RSB
                        00FD    ;
                        00FD    ; COULD NOT OPEN A FILE
                        00FD    ;
           FF00'    31  00FD    40$:  BRW     MAC$LAST_CHANCE             ; LAST CHANCE HANDLER
                        0100
```

```
                                   0100   285          .SBTTL  GET_OBJECT Obtain object file specification
                                   0100   286 ;++
                                   0100   287 ;
                                   0100   288 ; Functional Description:
                                   0100   289 ;
                                   0100   290 ;        Call the CLI to obtain the GLOBAL object file specification.
                                   0100   291 ;
                                   0100   292 ; Inputs:
                                   0100   293 ;
                                   0100   294 ;        4(AP)   adr     address of a dynamic string descriptor
                                   0100   295 ;                        (ultimately either a null string or the globally
                                   0100   296 ;                         specified value)
                                   0100   297 ;
                                   0100   298 ; Implicit Inputs:
                                   0100   299 ;
                                   0100   300 ;        LOC_OBJ_FLAG    specifies if /obj is locally present.
                                   0100   301 ;
                                   0100   302 ; Implicit Outputs:
                                   0100   303 ;
                                   0100   304 ;        If an object file specification was supplied, the address and length
                                   0100   305 ;        of the output file specification is set into the output file FAB.
                                   0100   306 ;
                                   0100   307 ; Side Effects:
                                   0100   308 ;
                                   0100   309 ;        1)  The address field of the dynamic descriptor is cleared
                                   0100   310 ;            (to force a new buffer to be allocated on the next invocation
                                   0100   311 ;            of the CLI).  This preserves the buffer address and its contents
                                   0100   312 ;            (the OBJECT file specification).
                                   0100   313 ;        2)  /LIST locally specified overrides global list specification.
                                   0100   314 ;
                                   0100   315 ;--
                             0000  0100   316          .ENTRY  GET_OBJECT,^M<>          ; Get the object file specification.
              0000008F'EF     9F  0102   317          PUSHAB  OBJECT                   ;
          00000000'GF   01    FB  0108   318          CALLS   #1, G^CLI$PRESENT        ; Was /OBJECT specified?
                    45  50    E9  010F   319          BLBC    R0, 20$                  ; No, Return.
          00000000'8F   50    D1  0112   320          CMPL    R0, #CLI$_LOCPRES        ; If locally present then fall thru
                        3C    13  0119   321          BEQL    20$                      ;   and handle in procedure GET_LOC_OBJ.
                                  011B   322
   00000000'8F    00000008'EF     D1  011B   323          CMPL    LOC_OBJ_FLAG, -          ; /NOOBJ positionally overrides
                                  0126   324                  #CLI$_LOCNEG             ;   global /OBJ
                        2F    13  0126   325          BEQL    20$                      ; ... Return
                  02    50    F0  0128   326          INSV    R0,#MAC_V_OBJECT,-       ; (Toggle /OBJECT flag.)
          00000000'GF   01        012B   327                  #1,G^MAC$GL_CTLMSK       ;
                  1D    01    F0  0131   328          INSV    #1,#FAB$V_OFP,-
          00000004'EF   01        0134   329                  #1,MAC$OBJECT_FAB+FAB$L_FOP
                                  013A   330                                           ; Set OFP bit - Use only portions
                                  013A   331                                           ;         of related filename.
                    04  AC    DD  013A   332          PUSHL   4(AP)                    ; Push the descriptor address.
              0000008F'EF     9F  013D   333          PUSHAB  OBJECT                   ; Push 'OBJECT' qualifier descriptor.
          00000000'GF   02    FB  0143   334          CALLS   #2, G^CLI$GET_VALUE      ; Get the file specification.
                    0A  50    E9  014A   335          BLBC    R0, 20$                  ; ...exit... use first input file
                    04  AC    DD  014D   336          PUSHL   4(AP)                    ; Push the string desc address.
          00000737'EF   01    FB  0150   337          CALLS   #1, OBJ_FILE             ; Set up the OBJECT file FAB.
              50    04  AC    D0  0157   338 20$:     MOVL    4(AP), R0                ; Get the address of the descriptor
                    04  A0    D4  015B   339          CLRL    DSC$A_POINTER (R0)       ; and reset it to force a new buffer.
                        60    B4  015E   340          CLRW    DSC$W_LENGTH (R0)        ; Also reset length for next segment.
                              04  0160   341          RET                              ; Return to continue the parse.
```

```
                                   0161           .SBTTL  GET_LISTING Obtain the listing file spec (et. al.)
                                   0161     ;++
                                   0161     ;
                                   0161     ; Functional Description:
                                   0161     ;
                                   0161     ;       Call the CLI to obtain the global list file specification
                                   0161     ;       and all the associated listing control qualifiers.
                                   0161     ;
                                   0161     ; Inputs:
                                   0161     ;
                                   0161     ;       4(AP)   adr     address of a dynamic string descriptor
                                   0161     ;
                                   0161     ; Implicit Inputs:
                                   0161     ;
                                   0161     ;       LOC_LST_FLAG    specifies if /list is locally present.
                                   0161     ;
                                   0161     ; Implicit Outputs:
                                   0161     ;
                                   0161     ;       If /LISTING was supplied, the /CROSS_REFERENCE and /SHOW options
                                   0161     ;       are selected either explicitly or by default action.
                                   0161     ;
                                   0161     ; Side Effects:
                                   0161     ;
                                   0161     ;       1) The address field of the dynamic descriptor is cleared.  This
                                   0161     ;          preserves the buffer address and its contents (the LISTING file
                                   0161     ;          specification).
                                   0161     ;       2) If a LISTING is specified both locally and globally then thE
                                   0161     ;          local listing is produced.
                                   0161     ;--
                             0000  0161           .ENTRY  GET_LISTING, ^M<>       ; Get the listing file specification.
          00000072'EF    9F        0163           PUSHAB  LISTING
       00000000'GF    01    FB     0169           CALLS   #1, G^CLI$PRESENT       ; See if /LISTING was specified.
                   61 50    E9     0170           BLBC    R0,20$                  ; Then ...Return.
       00000000'8F    50    D1     0173           CMPL    R0, #CLI$_LOCPRES       ; If locally present then fall thru
                      4F    13     017A           BEQL    10$                     ;   and handle GLOBAL list qualifiers
                                   017C
00000000'8F   00000004'EF   D1     017C           CMPL    LOC_LST_FLAG, -         ; /NOLIST positionally overrides
                                   0187                   #CLI$_LOCNEG            ;   global /LIST
                      4B    13     0187           BEQL    20$                     ; ... Return
                01    50    F0     0189           INSV    R0,#MAC_V_LIST,-        ; (Toggle /LISTING flag.)
       00000000'GF    01           018C                   #1,G^MAC$GL_CTLMSK      ;
                1D    01    F0     0192           INSV    #1,#FAB$V_OFP,-
       00000004'EF    01           0195                   #1,MAC$OBJECT_FAB+FAB$L_FOP
                                   019B                                           ; Set OFP bit - Use only portions
                                   019B                                           ;       of related filename.
                   04 AC    DD     019B           PUSHL   4(AP)                   ; Push the descriptor address.
          00000072'EF    9F        019E           PUSHAB  LISTING                 ; Push 'LISTING' qualifier descriptor.
       00000000'GF    02    FB     01A4           CALLS   #2, G^CLI$GET_VALUE     ; Get the file specification
                                   01AB                                           ;       or null string.
                   0A 50    E9     01AB           BLBC    R0, 5$                  ; ...exit... use first input file
                                   01AE
                   04 AC    DD     01AE           PUSHL   4(AP)                   ; Push the default file specification
                                   01B1                                           ;       as the string desc address.
       00000715'EF    01    FB     01B1           CALLS   #1, LIST_FILE           ; Set up the LISTING file FAB.
                                   01B8                                           ;
            50     04 AC    D0     01B8     5$:   MOVL    4(AP), R0               ; Get the address of the descriptor
                   04 A0    D4     01BC           CLRL    DSC$A_POINTER (R0)      ; and reset it to force a new buffer.
                   60   B4  01BF          CLRW    DSC$W_LENGTH (R0)        ; Also reset length for next segment.
                04 AC   DD  01C1          PUSHL   4(AP)                    ;
   00000304'EF     01   FB  01C4          CALLS   #1, LIST_QUALS           ; get the rest of the qualifiers.
                            01CB
          50    04 AC   D0  01CB    10$:  MOVL    4(AP), R0                ; Get the address of the descriptor
                04 A0   D4  01CF          CLRL    DSC$A_POINTER (R0)       ; and reset it to force a new buffer.
                   60   B4  01D2          CLRW    DSC$W_LENGTH (R0)        ; Also reset length for next segment.
                        04  01D4    20$:  RET
                            01D5
```

```
                                01D5          .SBTTL  GET_LOC_OBJECT Obtain the positional object file spec (et. al.)
                                01D5  ;++
                                01D5  ;
                                01D5  ; Functional Description:
                                01D5  ;
                                01D5  ;       This routine obtains the positional object file specification and
                                01D5  ;       and sets up the OFP bit in FAB FOP for object file related name.
                                01D5  ;
                                01D5  ; Inputs:
                                01D5  ;
                                01D5  ;       4(AP)   adr      address of a dynamic string descriptor (if object
                                01D5  ;                            file specified locally)
                                01D5  ;
                                01D5  ; Implicit Outputs:
                                01D5  ;
                                01D5  ;       LOC_OBJ_FLAG    Flag to hold status of CLI$PRESENT when qualifier
                                01D5  ;                         is locally present
                                01D5  ;
                                01D5  ; Side Effects:
                                01D5  ;
                                01D5  ;       1)  The address field of the dynamic descriptor is cleared.  This
                                01D5  ;           preserves the buffer address and its contents (the LISTING file
                                01D5  ;           specification).
                                01D5  ;       2)  If a OBJECT file is marked locally as /NOOBJECT then the object
                                01D5  ;           flag is reset here.
                                01D5  ;--
                         0054   01D5          .ENTRY  GET_LOC_OBJECT, ^M<R2,R4,R6>
        0000008F'EF        9F   01D7          PUSHAB  OBJECT                  ;
    00000000'GF     01     FB   01DD          CALLS   #1, G^CLI$PRESENT       ; See if /OBJECT was specified.
    00000000'8F     50     D1   01E4          CMPL    R0, #CLI$_LOCPRES       ; locally present
                    2B     13   01EB          BEQL    5$                      ;
    00000000'8F     50     D1   01ED          CMPL    R0, #CLI$_LOCNEG        ; locally negated
                                01F4                                          ;
                    63     12   01F4          BNEQ    20$                     ; ...exit...
                                01F6                                          ;
50    00000008'EF          D0   01F6          MOVL    LOC_OBJ_FLAG, R0        ; Check for prev file's
    00000000'8F     50     D1   01FD          CMPL    R0, #CLI$_LOCPRES       ;       positional /OBJ qual and
                                0204                                          ;       avoid clearing global flag
                    53     13   0204          BEQL    20$                     ; ...exit...
                                0206                                          ;
    00000008'EF     50     D0   0206          MOVL    R0, LOC_OBJ_FLAG        ; Save locally negative flag for later
             02     00     F0   020D          INSV    #0,#MAC_V_OBJECT,-      ; Clear object flag bit
    00000000'GF     01          0210                  #1,G^MAC$GL_CTLMSK
                    41     11   0216          BRB     20$                     ; ...exit...
                                0218                                          ;
    00000008'EF     50     D0   0218  5$:     MOVL    R0, LOC_OBJ_FLAG        ; Indicate that local object specified
             02     50     F0   021F          INSV    R0,#MAC_V_OBJECT,-      ; (Toggle /OBJECT flag to "ON".)
    00000000'GF     01          0222                  #1,G^MAC$GL_CTLMSK      ;
             1D     01     F0   0228          INSV    #1,#FAB$V_OFP,-
    00000004'EF     01          022B                  #1,MAC$OBJECT_FAB+FAB$L_FOP
                                0231                                          ; Set OFP bit - Use related filename.
                                0231                                          ;
                 04 AC     DD   0231          PUSHL   4(AP)                   ;
        C000008F'EF        9F   0234          PUSHAB  OBJECT                  ;
    00000000'GF     02     FB   023A          CALLS   #2, G^CLI$GET_VALUE     ; Get the file specification or null
                                0241                                          ;       string.
                 09 50     E8   0241          BLBS    R0, 15$                 ; Dont clear the OFP
```

```
          1D    00    F0  0244    467           INSV    #0,#FAB$V_OFP,-
00000004'EF     01        0247    468                   #1,MAC$OBJECT_FAB+FAB$L_FOP
                          024D    469                                                    ; Set OFP bit - Use related filename.
       04 AC    DD        024D    470 15$:      PUSHL   4(AP)                            ;
00000737'EF     01    FB  0250    471           CALLS   #1, OBJ_FILE                     ; Set up the OBJ file FAB specified.
                00    11  0257    472           BRB     20$                              ; ...exit...
                          0259    473                                                    ;
                          0259    474                                                    ;
     50    04 AC    D0    0259    475 20$:      MOVL    4(AP), R0                        ; Get the address of the descriptor
           04 A0    D4    025D    476           CLRL    DSC$A_POINTER (R0)               ; and reset it to force a new buffer.
              60    B4    0260    477           CLRW    DSC$W_LENGTH (R0)                ; Also reset length for next segment.
                    04    0262    478           RET
                          0263    479 ;
```

```
                                    0263          .SBTTL  GET_LOC_LIST Obtain the positional listing file spec (et. al.)
                                    0263     ;++
                                    0263     ;
                                    0263     ; Functional Description:
                                    0263     ;
                                    0263     ;        This routine obtains the positional listing file specification and
                                    0263     ;        sets up the related name for the listing file, and gets any qualifiers
                                    0263     ;        specified with the local listing.
                                    0263     ;
                                    0263     ; Inputs:
                                    0263     ;
                                    0263     ;        4(AP)    adr      address of a dynamic string descriptor (used if
                                    0263     ;                            specific list file is specified locally)
                                    0263     ;
                                    0263     ; Implicit Outputs:
                                    0263     ;
                                    0263     ;        LOC_LST_FLAG     Flag to hold status of CLI$PRESENT when qualifier
                                    0263     ;                            is locally present
                                    0263     ;
                                    0263     ;        If /LISTING was supplied, the /CROSS_REFERENCE and /SHOW options
                                    0263     ;        are selected either explicitly or by default action.
                                    0263     ;
                                    0263     ; Side Effects:
                                    0263     ;
                                    0263     ;        1)  The address field of the dynamic descriptor is cleared.  This
                                    0263     ;            preservcs the buffer address and its contents (the LISTING
                                    0263     ;            file specification).
                                    0263     ;        2)  If a LISTING file is marked locally as /NOLIST then the listing
                                    0263     ;            flag is reset here.
                                    0263     ;--
                              0054  0263          .ENTRY  GET_LOC_LIST,   ^M<R2,R4,R6>
         00000072'EF         9F     0265          PUSHAB  LISTING                  ;
 00000000'GF         01      FB     026B          CALLS   #1, G^CLI$PRESENT        ; See if /LISTING was specified.
 00000000'8F         50      D1     0272          CMPL    R0, #CLI$_LOCPRES        ; locally present
                     2B      13     0279          BEQL    5$                       ;
 00000000'8F         50      D1     027B          CMPL    R0, #CLI$_LOCNEG         ; locally negated
                     76      12     0282          BNEQ    20$                      ; ...exit...
                                    0284                                           ;
50  00000004'EF      D0             0284 3$:      MOVL    LOC_LST_FLAG, R0         ; Check for prev file's
 00000000'8F         50      D1     028B          CMPL    R0, #CLI$_LOCPRES        ;       positional /LIST qual and
                                    0292                                           ;       avoid clearing global flag
                     66      13     0292          BEQL    20$                      ; ...exit...
                                    0294                                           ;
 00000004'EF         50      D0     0294 4$:      MOVL    R0, LOC_LST_FLAG         ; Save locally negative flag for later
          01         00      F0     029B          INSV    #0,#MAC_V_LIST,-         ; Clear LIST flag bit
 00000000'GF         01             029E                  #1,G^MAC$GL_CTLMSK
                     54      11     02A4          BRB     20$                      ; ...exit...
                                    02A6                                           ;
 00000004'EF         50      D0     02A6 5$:      MOVL    R0, LOC_LST_FLAG         ; Indicate that local list specified
          1D         01      F0     02AD          INSV    #1,#FAB$V_OFP,-
 00000004'EF         01             02B0                  #1,MAC$LIST_FAB+FAB$L_FOP
                                    02B6                                           ; Set OFP bit - Use related filename.
          01         50      F0     02B6          INSV    R0,#MAC_V_LIST,-         ; (Toggle /LIST flag to "ON".)
 00000000'GF         01             02B9                  #1,G^MAC$GL_CTLMSK       ;
                                    02BF                                           ;
             04 AC           DD     02BF          PUSHL   4(AP)                    ;
      00000072'EF            9F     02C2          PUSHAB  LISTING                  ;
```

```
00000000'GF    02   FB  02C8    538          CALLS   #2, G^CLI$GET_VALUE     ; Get the file specification or null
                        02CF    539                                          ;        string.
          09 50     E8  02CF    540          BLBS    R0, 15$                 ; Dont clear the OFP
       1D   00      F0  02D2    541          INSV    #0,#FAB$V_OFP,-
00000004'EF  01         02D5    542                  #1,MAC$LIST_FAB+FAB$L_FOP
                        02DB    543                                          ; Set OFP bit - Use related filename.
          04 AC     DD  02DB    544 15$:     PUSHL   4(AP)                   ; Push the string descriptor address.
00000715'EF  01     FB  02DE    545          CALLS   #1, LIST_FILE           ; Set up the LIST file FAB specified.
             00     11  02E5    546          BRB     17$                     ; list_quals ...exit...
                        02E7    547                                          ;
     50    04 AC    D0  02E7    548 17$:     MOVL    4(AP), R0               ; Get the address of the descriptor
           04 A0    D4  02EB    549          CLRL    DSC$A_POINTER (R0)      ; and reset it to force a new buffer.
              60    B4  02EE    550          CLRW    DSC$W_LENGTH (R0)       ; Also reset length for next segment.
           04 AC    DD  02F0    551          PUSHL   4(AP)
00000304'EF  01     FB  02F3    552          CALLS   #1, LIST_QUALS          ; get the rest of the qualifiers.
                        02FA    553                                          ;
     50    04 AC    D0  02FA    554 20$:     MOVL    4(AP), R0               ; Get the address of the descriptor
           04 A0    D4  02FE    555          CLRL    DSC$A_POINTER (R0)      ; and reset it to force a new buffer.
              60    B4  0301    556          CLRW    DSC$W_LENGTH (R0)       ; Also reset length for next segment.
                    04  0303    557          RET
                        0304    558 ;
```

```
                                   0304   560         .SBTTL  LIST_QUALS Obtain the listing file associated qualifiers(et. al.)
                                   0304   561 ;++
                                   0304   562 ;
                                   0304   563 ; Functional Description:
                                   0304   564 ;
                                   0304   565 ;       This routine processes all the associated listing control qualifiers.
                                   0304   566 ;
                                   0304   567 ; Inputs:
                                   0304   568 ;
                                   0304   569 ;       4(AP)   adr     address of a dynamic string descriptor
                                   0304   570 ;
                                   0304   571 ; Implicit Outputs:
                                   0304   572 ;
                                   0304   573 ;       If /LISTING was supplied, the /CROSS_REFERENCE and /SHOW options
                                   0304   574 ;       are selected either explicitly or by default action.
                                   0304   575 ;
                                   0304   576 ; Side Effects:
                                   0304   577 ;
                                   0304   578 ;       None
                                   0304   579 ;
                                   0304   580 ;--
                             0000  0304   581         .ENTRY  LIST_QUALS, ^M<>                ; Get the listing file specification
        00000016'EF     9F         0306   582         PUSHAB  CROSS                   ; Push the CROSS_REFERENCE desc.
     00000000'GF    01  FB         030C   583         CALLS   #1, G^CLI$PRESENT       ; Was /CROSS_REFERENCE specified?
                 13 50  E9         0313   584         BLBC    R0, 15$                 ; No, skip ahead.
             04     50  F0         0316   585         INSV    R0,#MAC_V_CROSSREF,-    ; Toggle /CROSS_REFERENCE flag.
     00000000'GF    01             0319   586                 #1,G^MAC$GL_CTLMSK      ;
                 04 AC  DD         031F   587         PUSHL   4(AP)                   ; Yes, push the address of string descriptor
     00000776'EF    01  FB         0322   588         CALLS   #1, MACR_CROS           ; and process the keywords seperately.
                                   0329   589
        0000009D'EF     9F         0329   590 15$:    PUSHAB  SHOW                    ; Push the SHOW gulaifier descriptor.
     00000000'GF    01  FB         032F   591         CALLS   #1, G^CLI$PRESENT       ; Was /SHOW specified?
                 0A 50  E9         0336   592         BLBC    R0, 20$                 ; No, go process the /NOSHOW qualifier.
                 04 AC  DD         0339   593         PUSHL   4(AP)                   ; Address of string descriptor to use...
     000007E9'EF    01  FB         033C   594         CALLS   #1, MACR_SHOW           ; Yes, so process the keywords seprately
                                   0343   595
        00000081'EF     9F         0343   596 20$:    PUSHAB  NOSHOW                  ; Push the NOSHOW gulaifier descriptor.
     00000000'GF    01  FB         0349   597         CALLS   #1, G^CLI$PRESENT       ; Was /NOSHOW specified?
                 0A 50  E9         0350   598         BLBC    R0, 30$                 ; No, return to complete parse.
                 04 AC  DD         0353   599         PUSHL   4(AP)                   ; Address of string descriptor to use...
     000007F7'EF    01  FB         0356   600         CALLS   #1, MACR_NOSH           ; Yes, parse /NOSHOW's keywords...
                                   035D   601
                        04         035D   602 30$:    RET                             ;
```

```
                                   035E          .SBTTL  GET_INPUT Obtain the input file specification(s)
                                   035E     ;++
                                   035E     ;
                                   035E     ; Functional Description:
                                   035E     ;
                                   035E     ;       The input file specification is obtained and all associated qualif'ers
                                   035E     ;       are processed.
                                   035E     ;
                                   035E     ; Inputs:
                                   035E     ;
                                   035E     ;       4(AP)   adr     address of a dynamic string descriptor
                                   035E     ;
                                   035E     ; Implicit Inputs:
                                   035E     ;       The default qualifiers defined in the .CLD are inputs to this routine.
                                   035E     ;
                                   035E     ; Implicit Outputs:
                                   035E     ;
                                   035E     ;       The input file specification is parsed and processed, and the qualifiers
                                   035E     ;       (/DEBUG, /DISABLE, /ENABLE) options have been set.
                                   035E     ;
                                   035E     ; Side Effects:
                                   035E     ;
                                   035E     ;       Allocation of STACK LOCAL Storage is used as follows:
                                   035E     ;
                                   035E     ;               +--------------------+
                                   035E     ;               |class|type| length  | 0(SP)
                                   035E     ;               +--------------------+
                                   035E     ;               |   string address   | 4(SP)
                                   035E     ;               +--------------------+
                                   035E     ;               |    saved status    | 8(SP)
                                   035E     ;               +--------------------+
                                   035E     ;
                                   035E     ;--
                             0000  035E          .ENTRY  GET_INPUT, ^M<>         ; Get the input file specification
                                   0360     ;
                                   0360     ; Initialization...
                                   0360     ;
            5E    F4 AE     9E     0360          MOVAB   -12(SP), SP             ; Allocate some LOCAL storage.
                     6E     7C     0364          CLRQ    (SP)                    ; Initialize a dynamic string
                     02     90     0366          MOVB    #DSC$K_CLASS_D,-        ; descriptor.  The remaining longword
                  03 AE            0368                  DSC$B_CLASS (SP)        ; is used to hold a 'saved status'.
                     50     D4     036A          CLRL    R0
         00000004'EF 50     D0     036C          MOVL    R0, LOC_LST_FLAG        ; Initialize positional flags
         00000008'EF 50     D0     0373          MOVL    R0, LOC_OBJ_FLAG        ;       for every concat file list.
             00000000'GF    D4     037A 10$:     CLRL    G^MAC$GL_DIRFLG         ; Reset the LIBRARY flag and
             00000000'EF    D4     0380          CLRL    UPDATE_LIST             ; the UPDATE flag (MUTEX)
00000000'GF  00000060 8F    CA     0386          BICL2   #<MAC_M_LIBRARY!MAC_M_UPDATE>, -
                                   0391                  G^MAC$GL_CTLMSK         ; /LIBRARY and /UPDATE are mutually
                                   0391                                          ; exclusive on a SINGLE file spec!
                                   0391                                          ;       ...(Reset qualifier bits)
                     5E     DD     0391          PUSHL   SP                      ; Push address of descriptor
             00000057'EF    9F     0393          PUSHAB  FILE                    ; Push the 'FILE' parameter descriptor.
         00000000'GF 02     FB     0399          CALLS   #2, G^CLI$GET_VALUE     ; Get the input file specification.
                  03 50     F0     03A0          INSV    RC,#MAC_V_INPFIL,-      ; Toggle INPUT file parsed flag.
         00000000'GF 01            03A3                  #1,G^MAC$GL_CTLMSK      ;
               08 AE 50     D0     03A9          MOVL    R0, 8(SP)               ; Save status...
                                   03AD                                          ;       check later for more files.
            03 50  E8  03AD          BLBS    R0, 20$                  ; Any more input files?
             0108  31  03B0          BRW     90$                      ; No, return and finish processing.
                       03B3      ;
                       03B3      ; Input file qualifier processing...
                       03B3      ;
      00000049'EF  9F  03B3  20$:    PUSHAB  ENABLE                   ; Push the ENABLE qualifier descriptor.
   00000000'GF  01 FB  03B9          CALLS   #1, G^CLI$PRESENT        ; Was /ENABLE specified?
            0A 50  E9  03C0          BLBC    R0, 30$                  ; No, check the next qualifier.
            04 AC  DD  03C3          PUSHL   4(AP)                    ; Use common dynamic string descriptor.
   00000812'EF  01 FB  03C6          CALLS   #1, MACR_ENAB            ; Yes, process the ENABLE keywords.
                       03CD
      0000003A'EF  9F  03CD  30$:    PUSHAB  DISABLE                  ; Push the DISABLE qualifier descriptor.
   00000000'GF  01 FB  03D3          CALLS   #1, G^CLI$PRESENT        ; Was /DISABLE specified?
            0A 50  E9  03DA          BLBC    R0, 40$                  ; No, check the next qualifier.
            04 AC  DD  03DD          PUSHL   4(AP)                    ; Use common dynamic string descriptor.
   00000820'EF  01 FB  03E0          CALLS   #1, MACR_DISA            ; Yes, process the DISABLE keywords.
                       03E7
      0000002D'EF  9F  03E7  40$:    PUSHAB  DEBUG                    ; Push the DEBUG qualifier descriptor.
   00000000'GF  01 FB  03ED          CALLS   #1, G^CLI$PRESENT        ; Was /DEBUG specified?
            0C 50  E9  03F4          BLBC    R0, 50$                  ; No, see if negations are performed.
            04 AC  DD  03F7          PUSHL   4(AP)                    ; Yes, use common dynamic string desc.
   000008CB'EF  01 FB  03FA          CALLS   #1, MACR_DEBU            ; and process the DEBUG keywords.
               13  11  0401          BRB     60$                      ; Done, go process the next qualifier.
   00000000'8F  50 D1  0403  50$:    CMPL    R0, #CLI$_NEGATED        ; Was /NODEBUG explicitly specified?
               0A  12  040A          BNEQ    60$                      ; No, skip over negation activities.
            04 AC  DD  040C          PUSHL   4(AP)                    ; Yes, use common dynamic string desc.
   000008BF'EF  01 FB  040F          CALLS   #1, MACR_NODE            ; and process negations.
                       0416
      000000A9'EF  9F  0416  60$:    PUSHAB  UPDATE                   ; Push the UPDATE qualifier descriptor.
   00000000'GF  01 FB  041C          CALLS   #1, G^CLI$PRESENT        ; Was /UPDATE specified?
            06  50 F0  0423          INSV    R0,#MAC_V_UPDATE,-       ; Toggle /UPDATE flag.
   00000000'GF  01     0426                  #1,G^MAC$GL_CTLMSK       ;
            0A 50  E9  042C          BLBC    R0, 65$                  ; No, process /LIBR, /OBJ and /LIST.
            04 AC  DD  042F          PUSHL   4(AP)                    ; Use common dynamic string descriptor.
   000009F4'EF  01 FB  0432          CALLS   #1, MACR_UPDA            ; Yes, set up UPDATE file list.
                       0439      ;
                       0439      ;  Process global library, listing & object qualifiers...
                       0439      ;
      00000063'EF  9F  0439  65$:    PUSHAB  LIBRARY                  ; Push the LIBRARY qualifier descriptor.
   00000000'GF  01 FB  043F          CALLS   #1, G^CLI$PRESENT        ; Is this file a LIBRARY?
            05  50 F0  0446          INSV    R0,#MAC_V_LIBRARY,-      ; Toggle /LIBRARY flag.
   00000000'GF  01     0449                  #1,G^MAC$GL_CTLMSK       ;
00000000'GF  01  00  50  F0  044F    INSV    R0,#0,#1,G^MAC$GL_DIRFLG; Set TRUE or FALSE in the FLAGS.
                       0458                                           ;
   00000000'8F  50 D1  0458          CMPL    R0, #CLI$_LOCPRES        ; /LIBRARY positionally specified
               1E  13  045F          BEQL    75$                      ; then skip global file processing
                       0461                                           ;
      08 00000008'EF  E8  0461       BLBS    LOC_OBJ_FLAG, 70$        ; Skip if prev local obj spec
            04 AC  DD  0468          PUSHL   4(AP)                    ; Process the OBJECT file specification
        FC90 CF  01 FB  046B         CALLS   #1, GET_OBJECT           ; (/OBJECT).
      08 00000004'EF  E8  0470  70$: BLBS    LOC_LST_FLAG, 75$        ; Skip if prev local list spec
            04 AC  DD  0477          PUSHL   4(AP)                    ; Process the LISTING file specification
        FCE2 CF  01 FB  047A         CALLS   #1, GET_LISTING          ; (/LISTING, /CROSS_REFERENCE and /SHOW)
                       047F
               5E  DD  047F  75$:    PUSHL   SP                       ; Now, process the input file...
   000004BC'EF  01 FB  0481          CALLS   #1, INP_FILE             ; (source or library file).
                       0488      ;
```

```
                                0488      ; Process positional listing/object qualifiers
                                0488      ;
                04 AC     DD    0488            PUSHL   4(AP)                   ; Process the OBJECT file specification
        FD45 CF    01     FB    048B            CALLS   #1, GET_LOC_OBJECT      ; (/OBJECT).
                04 AC     DD    0490            PUSHL   4(AP)                   ; Process any new LISTING file
                                0493                                            ;       specification
        FDCB CF    01     FB    0493            CALLS   #1, GET_LOC_LIST        ; Local /LISTING, /CROSS_REFERENCE
                                0498                                            ;       and /SHOW.
                                0498      ;
                                0498      ; Process cleanup...
                                0498      ;
                50    5E  D0    0498            MOVL    SP, R0                  ; Get the address of the descriptor
                04    A0  D4    049B            CLRL    DSC$A_POINTER (R0)      ; and reset it to force a new buffer.
                      60  B4    049E            CLRW    DSC$W_LENGTH (R0)       ; Also reset length for next segment.
                                04A0
   08 AE    00000000'8F   D1    04A0            CMPL    #CLI$_CONCAT, 8(SP)     ; Is this a corcatinated input file?
                      03  12    04A8            BNEQ    80$                     ; No, but there may be more input files
                    FECD  31    04AA            BRW     10$                     ; Yes, continue processing the
                                04AD                                            ;       next file.
   08 AE    00000000'8F   D1    04AD      80$:  CMPL    #CLI$_COMMA, 8(SP)      ; Check to see if there really are any
                      04  12    04B5            BNEQ    90$                     ; more input file(s).
            00 6B     23  E2    04B7            BBSS    #FLG$V_MOREINP,(R11),90$; Indicate more input file(s) expected.
                          04    04BB      90$:  RET                             ; Return, to complete current command.
                                04BC
```

```
                                    04BC          .SBTTL  ACTION ROUTINE FOR INPUT FILE SPECIFICATION
                                    04BC
                                    04BC  ;++
                                    04BC  ; FUNCTIONAL DESCRIPTION
                                    04BC  ;
                                    04BC  ;       THIS ROUTINE CREATES A FAB AND LINKS IT INTO THE LIST OF
                                    04BC  ;       INPUT FABS.  IT THEN OPENS THE FILE TO MAKE SURE IT EXISTS
                                    04BC  ;       BEFORE EXITING.
                                    04BC  ;
                                    04BC  ;--
                              00FC  04BC          .ENTRY  INP_FILE, ^M<R2, R3, R4, R5, R6, R7>
                                    04BE                                          ;HANDLE INPUT FILE SPECIFICATION
           03 00000000'EF      E9   04BE          BLBC    L^MAC$GL_DIRFLG,10$     ;BRANCH IF NOT LIBRARY FILE
                     017D      31   04C5          BRW     INP_LIBR_FILE           ;ELSE GO HANDLE LIBRARY SPECIALLY
                                    04C8          ;
                                    04C8          ; CREATE INPUT FILE DESCR. BLOCK
                                    04C8          ;
 00000000'GF   00000000'EF     FA   04C8  10$:    CALLG   L^MAC$G_1_PAGE,G^LIB$GET_VM
                                    04D3                                          ;ALLOCATE AN FDB
                    03 50      E8   04D3          BLBS    R0,20$                  ;BR IF SUCCESS
                     0144      31   04D6          BRW     100$                    ;NO--GO REPORT ERROR
           56  00000000'GF     D0   04D9  20$:    MOVL    G^MAC$GL_BASEADDR,R6    ;GET BLOCK ADDRESS
 66   0200 8F   00   6E   00   2C   04E0          MOVC5   #0,(SP),#0,#512,(R6)    ;CLEAR THE BLOCK
                    56 08      C0   04E8          ADDL2   #8,R6                   ;POINT TO FDB (SKIP THE LINKS)
                  5003 8F      B0   04EB          MOVW    #<<FAB$C_BLN>@8+FAB$C_BID>,-
                       66           04EF                          FAB$B_BID(R6)   ;IDENTIFY AS A FAB WITH IDENT AND LENGTH
                16 A6  02      88   04F0          BISB2   #FAB$M_GET,FAB$B_FAC(R6);SET FOR GETS
         35 A6  00000004'EF    90   04F4          MOVB    L^INP_DEFEXT,FAB$B_DNS(R6)
                                    04FC                                          ;SET LENGTH OF DEFAULT NAME STRING
         30 A6  00000005'EF    9E   04FC          MOVAB   L^INP_DEFEXT+1,FAB$L_DNA(R6)
                                    0504                                          ;AND ITS ADDRESS
                00000000'EF    9E   0504          MOVAB   L^MAC$INPUT_XAB,-       ;STORE ADDRESS OF XAB FOR DATE
                      24 A6         050A                          FAB$L_XAB(R6)
                                    050C          ;
                                    050C          ; NAM BLOCK FOR SUM$xxxx ROUTINES USED THROUGHOUT MACRO
                                    050C          ;
                51   50 A6     9E   050C          MOVAB   FAB$C_BLN(R6),R1        ;POINT TO NAM BLOCK
                28 A6   51     D0   0510          MOVL    R1,FAB$L_NAM(R6)        ;STORE NAM ADDRESS IN FAB
             02 A1   FF 8F     90   0514          MOVB    #NAM$C_MAXRSS,NAM$B_RSS(R1)
                                    0519                                          ;STORE IN NAM BLOCK
         04 A1  00000000'EF    9E   0519          MOVAB   L^MAC$INP_NAM_BUF,NAM$L_RSA(R1)
                                    0521                                          ;AND ITS ADDRESS
                  6002 8F      B0   0521          MOVW    #<<NAM$C_BLN>@8+NAM$C_BID>,-
                    50 A6           0525                          FAB$C_BLN+NAM$B_BID(R6)
                                    0527                                          ;IDENTIFY NAM BLOCK AS A NAM BLOCK
         10 A1  00000000'EF    9E   0527          MOVAB   L^MAC$INPUT_RLFNM,NAM$L_RLF(R1)
                                    052F                                          ;SET RELATED NAM BLOCK ADDRESS
         0C A1  00000000'EF    9E   052F          MOVAB   L^MAC$INP_NAM_BUF,NAM$L_ESA(R1)
                                    0537                                          ;SET EXPANDED STRING BUFFER ADDRESS
             0A A1   FF 8F     90   0537          MOVB    #NAM$C_MAXRSS,NAM$B_ESS(R1)
                                    053C                                          ;  AND ITS SIZE
                                    053C          ;
                                    053C          ; USE LIB$FIND_FILE TO GET THE INPUT FILE NAME
                                    053C          ;
                       7E      7C   053C          CLRQ    -(SP)                   ; Initialize a dynamic string
                       02      90   053E          MOVB    #DSC$K_CLASS_D,-        ; descriptor.
                    03 AE           0540                  DSC$B_CLASS (SP)        ;
```

```
                      0E    90  0542           MOVB    #DSC$K_DTYPE_T,-          ;SETUP THE TYPE
                   02 AE        0544                   DSC$B_DTYPE (SP)
                57    5E    D0  0546           MOVL    SP, R7                    ; RESULT WILL CONTAIN THE RESULT DESC
                                0549                                             ;
           00000000'EF      DF  0549           PUSHAL  L^MAC$GC_FFLAGS           ;USER FLAGS
           00000000'EF      DF  054F           PUSHAL  L^MAC$GL_STV_ADDR         ;STV_ADDR
                    00      DD  0555           PUSHL   #0                        ;RELATED NAME(zero - not used)
           00000000'EF      9F  0557           PUSHAB  L^MAC$INP_DEFNAM          ;DEFAULT NAME (filename extension)
           00000000'EF      DF  055D           PUSHAL  L^MAC$GL_CONTEXT          ;CONTEXT (of previous call)
                    57      DD  0563           PUSHL   R7                        ;RESULT DESC ADDRESS
                 04 AC      DD  0565           PUSHL   4(AP)                     ;FILENAME DESC ADDRESS
      00000000'GF   07      FB  0568           CALLS   #7, G^LIB$FIND_FILE       ;SEARCH FOR FILENAME
                                056F                                             ;
                 03 50      E8  056F           BLBS    R0,30$
                  00B2      31  0572           BRW     120$                      ;ERROR - REPORT UNABLE TO FIND
                                0575           ;
                                0575           ;  PERFORM THE REQUIRED BOOKKEEPING
                                0575           ;
                 04 A7      D0  0575  30$:     MOVL    DSC$A_POINTER(R7),-       ;STORE ADDRESS OF FILENAME STRING
                 2C A6          0578                           FAB$L_FNA(R6)     ; IN THE FAB
                    67      90  057A           MOVB    DSC$W_LENGTH(R7),-        ;STORE LENGTH OF FILENAME STRING
                 34 A6          057C                           FAB$B_FNS(R6)     ; IN THE FAB
                                057E
                                057E           $PARSE  FAB=(R6),-                ;OPEN THE INPUT FILE TO ENSURE EXISTENCE
                                057E                   ERR=W^MAC$ERR_OPN_INP
                 03 50      E8  058B           BLBS    R0,35$
                  0093      31  058E           BRW     115$                      ;BR IF THERE WAS AN ERROR
                                0591
          51     50 A6      9E  0591  35$:     MOVAB   FAB$C_BLN(R6),R1          ;POINT TO NAM BLOCK
                                0595           ;
                                0595           ; SET LENGTHS FOR SUM$xxx ROUTINES
                                0595           ;
                                0595
                    67      90  0595           MOVB    DSC$W_LENGTH(R7),-
           0000000B'EF          0597                   L^MAC$INPUT_RLFNM+NAM$B_ESL
                                059C                                             ;COPY LENGTH INTO EXPANDED STRING
                    67      90  059C           MOVB    DSC$W_LENGTH(R7),-
           00000003'EF          059E                   L^MAC$INPUT_RLFNM+NAM$B_RSL
                                05A3                                             ;COPY LENGTH AND
                                05A3                                             ; CONTENTS
                    67      90  05A3           MOVB    DSC$W_LENGTH(R7),-
                 03 A1          05A5                   NAM$B_RSL(R1)
                                05A7
 00000000'EF   04 B7    67  28  05A7           MOVC3   DSC$W_LENGTH(R7), @DSC$A_POINTER(R7), -
                                05B0                   L^MAC$INP_NAM_BUF
                                05B0
                                05B0
     00000004'FF     F8 A6  0E  05B0           INSQUE  -8(R6),@L^MAC$GL_INPQUE+4
                                05B8                                             ;INSERT THE FDB INTO THE INPUT QUE
                                05B8           ;
                                05B8           ;  SAVE RESULT STRING FOR LISTING AND OBJECT FILES
                                05B8           ;  WHICH ARE SPECIFIED POSITIONALLY
                                05B8           ;
       52      00000000'EF  DE  05B8           MOVAL   L^MAC$GL_RESULT, R2       ;SAVE THE RESULT NAME STRING
                    67      B0  05BF           MOVW    DSC$W_LENGTH(R7),-
                    62          05C1                    DSC$W_LENGTH(R2)         ;COPY LENGTH OF EXPANDED STRING
                                05C2                                             ; FOR OBJECT AND LISTING FILES
 00000000'EF    04 B7    67    28  05C2   857          MOVC3    DSC$W_LENGTH(R7), @DSC$A_POINTER(R7), -
                                   05CB   858                   L^MAC$GL_RESADDR         ;COPY EXPANDED NAME STRING
                                   05CB   859          ;
                                   05CB   860          ;
                                   05CB   861                                            ;FOR OBJECT AND LISTING FILES
        26 00000000'GF         E8  05CB   862          BLBS     G^MAC$GL_FINPTF, 40$     ;BRANCH IF NOT FIRST INPUT FILE
           00000000'GF         D6  05D2   863          INCL     G^MAC$GL_FINPTF          ;FLAG FIRST INPUT FILE DONE
                                   05D8   864          ;
                                   05D8   865          ;
                                   05D8   866          ;  SAVE THE NAME FOR DEFAULT OBJECT FILENAME DETERMINATION
                                   05D8   867          ;
                         67    90  05D8   868          MOVB     DSC$W_LENGTH(R7),-
           00000003'EF             05DA   869                   L^MAC$OBJPT_RLFNM+NAM$B_RSL
                                   05DF   870                                            ;COPY LENGTH OF EXPANDED STRING
 00000000'EF    04 B7    67    28  05DF   871          MOVC3    DSC$W_LENGTH(R7), @DSC$A_POINTER(R7), -
                                   05E8   872                   L^MAC$OB._NAM_BUF        ;COPY EXPANDED NAME STRING
                                   05E8   873                                            ;FOR OBJECT AND LISTING FILES
                         67    90  05E8   874          MOVB     DSC$W_LENGTH(R7),-
           00000003'EF             05EA   875                    L^MAC$LISPT_RLFNM+NAM$B_RSL
                                   05EF   876                                            ;COPY LENGTH OF EXPANDED STRING
 00000000'EF    04 B7    67    28  05EF   877          MOVC3    DSC$W_LENGTH(R7), @DSC$A_POINTER(R7), -
                                   05F8   878                   L^MAC$LIS_NAM_BUF        ;COPY EXPANDED NAME STRINC
                                   05F8   879                                            ;
      04 A6   01000000 8F      C8  05F8   880 40$:     BISL2    #FAB$M_NAM, FAB$L_FOP(R6)
                                   0600   881                                            ;SET TO OPEN BY NAM BLOCK
    00B0 C6   00000000'EF      D0  0600   882          MOVL     L^UPDATE_LIST, -         ;Store update files list
                                   0609   883                   FAB$C_BLN+NAM$C_BLN(R6)
                         11    13  0609   884          BEQL     60$                      ;If EQL there is no list
                    00B0 C6    DF  060B   885          PUSHAL   FAB$C_BLN+NAM$C_BLN(R6) ;Update list
                      50 A6    DF  060F   886          PUSHAL   FAB$C_BLN(R6)            ;NAM block address
       00000000'GF       02    FB  0612   887          CALLS    #2,G^SUM$OPEN            ;Open update files
                      08 50    E9  0619   888          BLBC     R0,115$                  ;Error if LBC
                               04  061C   889 60$:     RET
                                   061D   890          ;
                                   061D   891          ; COULD NOT ALLOCATE AN FDB
                                   061D   892          ;
       00000000'GF       00    FB  061D   893 100$:    CALLS    #0,G^MAC$ERR_NOMEM_0     ;REPORT ERROR
                      F9D9'    31  0624   894 115$:    BRW      MAC$LAST_CHANCE          ;GO TO LAST CHANCE HANDLER
                                   0627   895          ;
                                   0627   896          ; COULD NOT OPEN INPUT FILE
                                   0627   897          ;
                         50    DD  0627   898 120$:    PUSHL    R0                       ;SAVE RMS STS
           00000000'EF         DF  0629   899          PUSHAL   L^MAC$GL_STV_ADDR        ;SAVE THE STATUS VALUE ADDRESS
                         57    DD  062F   900          PUSHL    R7                       ;SAVE THE RESULT ADDRESS AND
       00000000'GF       03    FB  0631   901          CALLS    #3,G^MAC$ERR_FIND_INP    ;REPORT ERROR
                      F9C5'    31  0638   902          BRW      MAC$LAST_CHANCE          ;GO TO LAST CHANCE HANDLER
                                   063B   903
                               063B          .SBTTL  PROCESS LIBRARY FILE SPECIFIER
                               063B
                               063B  ;++
                               063B  ; FUNCTIONAL DESCRIPTION:
                               063B  ;
                               063B  ;       THIS ROUTINE IS CALLED WHEN THE /LIBRARY QUALIFIER IS SEEN
                               063B  ;       IN A FILE SPECIFICATION.  ALL THAT IS DONE IS TO SET THE
                               063B  ;       LOW BIT IN MAC$GL_DIRFLG.  THIS WILL CAUSE THE INP_FILE
                               063B  ;       ROUTINE TO TRANSFER CONTROL TO INP_LIBR_FILE WHEN IT IS
                               063B  ;       EVENTUALLY CALLED.
                               063B  ;--
                               063B
                               063B  LIBR_FILE:
                         0000  063B          .WORD   0                       ;REGISTER SAVE MASK
      00000000'EF    01    88  063D          BISB    #1,L^MAC$GL_DIRFLG      ;SET LIBRARY FILE FLAG FOR INP_FILE
                           04  0644          RET
                               0645
                               0645  ;++
                               0645  ; FUNCTIONAL DESCRIPTION:
                               0645  ;
                               0645  ;       THIS ROUTINE IS CALLED WHEN THE INPUT SPECIFICATION IS COMPLETE
                               0645  ;       AND THE SPECIFICATION HAS /LIBRARY ON IT.  THE LIBRARY IS OPENED,
                               0645  ;       AND AN IN-CORE INDEX IS CREATED.  THIS ROUTINE USES LIB$FIND_FILE
                               0645  ;       WITH THE CURRENT GLOBAL CONTEXT.  THIS ALLOWS STICKINESS
                               0645  ;       WITH OTHER FILES ON THE COMMAND LINE.
                               0645  ;
                               0645  ;
                               0645  ;--
                               0645
                               0645  INP_LIBR_FILE:
                               0645                                          ;NO SAVE MASK SINCE WE BRW HERE
                               0645                                          ;FROM INP_FILE
                  F9B8'    30  0645          BSBW    MAC$ALL_MLB_MLF         ; Allocate and initialise MLF block
                               0648          ;
                               0648          ; USE LIB$FIND_FILE TO GET THE INPUT FILE NAME (Use same context as
                               0648          ;       as that used in Routine INP_FILE )
                               0648          ;
                     7E    7C  0648          CLRQ    -(SP)                   ; Initialize a dynamic string
                     02    90  064A          MOVB    #DSC$K_CLASS_D,-        ; descriptor.
                  03 AE        064C                  DSC$B_CLASS (SP)        ;
                     0E    90  064E          MOVB    #DSC$K_DTYPE_T,-        ;SETUP THE TYPE
                  02 AE        0650                  DSC$B_DTYPE (SP)
               57    5E    D0  0652          MOVL    SP, R7                  ; RESULT WILL CONTAIN THE RESULT DESC
                               0655                                          ;
         00000000'EF       DF  0655          PUSHAL  L^MAC$GC_FFLAGS         ;USER FLAGS
         00000000'EF       DF  065B          PUSHAL  L^MAC$GL_STV_ADDR       ;STV ADDR
                     00    DD  0661          PUSHL   #0                      ;RELATED NAME(zero - not used)
         00000000'EF       9F  0663          PUSHAB  L^MAC$MLB_DEFNAM        ;DEFAULT NAME (filename extension)
         00000000'EF       DF  0669          PUSHAL  L^MAC$GL_CONTEXT        ;CONTEXT (of previous call)
                     57    DD  066F          PUSHL   R7                      ;RESULT DESC ADDRESS
                  04 AC    DD  0671          PUSHL   4(AP)                   ;FILENAME DESC ADDRESS
      00000000'GF    07    FB  0674          CALLS   #7, G^LIB$FIND_FILE     ;SEARCH FOR FILENAME
                               067B                                          ;
                  03 50    E8  067B          BLBS    R0,5$
                   0080    31  067E          BRW     20$                     ;ERROR - REPORT UNABLE TO FIND
                               0681  ;
                               0681  ; Call librarian procedure to initialise library
```

```
                              0681   ;
             18 A6    9F      0681   5$:    PUSHAB  MLF$X_NAMBLK(R6)          ; Address of NAM block
        00000000'EF   9F      0684          PUSHAB  L^MAC$GL_LIBTYPE          ; Address of type = MLB
        00000000'EF   9F      068A          PUSHAB  L^MAC$GL_LIBFUNC          ; Address of function = READ
             14 A6    9F      0690          PUSHAB  MLF$L_CTINDEX(R6)         ; Address of control table index
    00000000'GF   04  FB      0693          CALLS   #4,G^LBR$INI_CONTROL      ; Initialise control block
             58 50    E9      069A          BLBC    R0,10$                    ; If LBC error
                              069D   ;
                              069D   ; Call librarian procedure to open library file
                              069D   ;
             0C A6    9F      069D          PUSHAB  MLF$Q_FNAMDS(R6)          ; Address of resultant length
             0C A6    9F      06A0          PUSHAB  MLF$Q_FNAMDS(R6)          ; Address of resultant descriptor
        00000000 EF   9F      06A3          PUSHAB  L^MAC$INPUT_RLFNM         ; Address of related file NAM block
        00000000'EF   9F      06A9          PUSHAB  L^MAC$MLB_DEFNAM          ; Address of default file name descriptor
                00    DD      06AF          PUSHL   #0                        ; Create options = 0
                67    9F      06B1          PUSHAB  DSC$W_LENGTH(R7)          ; Address of file name descriptor
             14 A6    9F      06B3          PUSHAB  MLF$L_CTINDEX(R6)         ; Address of control table index
    00000000'GF   07  FB      06B6          CALLS   #7,G^LBR$OPEN             ; Open library file
             35 50    E9      06BD          BLBC    R0,10$                    ; Error if LBC
             78 A6    9E      06C0          MOVAB   MLF$T_FNAM(R6),-          ; Set up related name NAM block to
        00000004'EF           06C3                  MAC$INPUT_RLFNM+NAM$L_RSA ; point to current expanded file name
             23 A6    90      06C8          MOVB    MLF$X_NAMBLK+NAM$B_ESL(R6),- ; Set up related file name block
        00000003'EF           06CB                  L^MAC$INPUT_RLFNM+NAM$B_RSL  ; to point to currect expanded name
    00000000'EF   66  0E      06D0          INSQUE  (R6),L^MAC$GL_MLB_QUE     ;INSERT AT FRONT OF MLB QUEUE
        00000000'GF   D6      06D7          INCL    G^MAC$GL_MLB_CNT          ; Increment macro library count
                              06DD                                            ;
                              06DD          ;
                              06DD          ;  SAVE RESULT STRING FOR LISTING AND OBJECT FILES
                              06DD          ;  WHICH ARE SPECIFIED POSITIONALLY
                              06DD          ;
          57   0C A6  DE      06DD          MOVAL   MLF$Q_FNAMDS(R6), R7      ;GET ADDR OF THE RESULT DESCRIPTOR
    52  00000000'EF   DE      06E1          MOVAL   L^MAC$GL_RESULT, R2       ;SAVE IN THE GLOBAL RESULT NAME STRING
             62   67  90      06E8          MOVB    (R7), DSC$W_LENGTH(R2)    ;COPY LENGTH OF EXPANDED STRING
                              06EB                                            ; FOR OBJECT AND LISTING FILES
00000000'EF   04 B7   67  28  06EB          MOVC3   DSC$W_LENGTH(R7), @DSC$A_POINTER(R7), -
                              06F4                  L^MAC$GL_RESADDR          ;COPY EXPANDED NAME STRING
                              06F4                                            ; FOR OBJECT AND LISTING FILES
                              06F4                                            ;
                      04      06F4          RET                               ;RETURN TO CLI
                              06F5   ;
                              06F5   ; LIBRARY OPEN ERROR
                              06F5   ;
                              06F5   10$:
                56    DD      06F5          PUSHL   R6                        ; Address of MLF
    00000000'EF   01  FB      06F7          CALLS   #1,MAC$ERR_LBROPEN        ; Report error
             F8FF'    31      06FE          BRW     MAC$LAST_CHANCE
                              0701
                              0701          ;
                              0701          ; COULD NOT OPEN INPUT FILE
                              0701          ;
                50    DD      0701   20$:   PUSHL   R0                        ;SAVE RMS STS
        00000000'EF   DF      0703          PUSHAL  L^MAC$GL_STV_ADDR         ;SAVE THE STATUS VALUE ADDRESS
                57    DD      0709          PUSHL   R7                        ;SAVE THE RESULT ADDRESS AND
    00000000'GF   03  FB      070B          CALLS   #3,G^MAC$ERR_FIND_INP     ;REPORT ERROR
             F8EB'    31      0712          BRW     MAC$LAST_CHANCE           ;GO TO LAST CHANCE HANDLER
```

```
                                    0715           .SBTTL  ACTION ROUTINE FOR OBJECT AND LISTING FILES
                                    0715
                                    0715   ;++
                                    0715   ; FUNCTIONAL DESCRIPTION:
                                    0715   ;
                                    0715   ;       THESE ROUTINES, LIST_FILE AND OBJ_FILE, ARE CALLED BY
                                    0715   ;       GET_LISTING, GET_OBJECT, GET_LOC_LIST, GET_LOC_OBJECT
                                    0715   ;       WHEN THE QUALFIERS ARE FOUND IN A COMMAND LINE.
                                    0715   ;       THE INFORMATION IS FILLED INTO THE PROPER FAB.
                                    0715   ;
                                    0715   ; Side Effects:
                                    0715   ;       THE CURRENT RESULTANT NAME (is stored in a global descriptor) IS
                                    0715   ;       INSERTED INTO THE RELATED FILENAME BLOCK.  THIS IS DONE HERE
                                    0715   ;       BECAUSE THIS IS THE "ONLY" PLACE THAT HANDLES LIST FILES AND
                                    0/15   ;       OBJECT FILES AFTER THE RESULTANT NAME IS COMPLETE FOR POSITIONAL
                                    0715   ;       QUALIFIERS.  ALSO THE IFI FEILD IS CLEARED.
                                    0715   ;
                                    0715   ;--
                                    0715
                                    0715   LIST_FILE:
                              0000  0715           .WORD   0                       ;REGISTER SAVE MASK
                                    0717           ;
                                    0717           ; INSERT THE RESULTANT FILENAME INTO THE RELATED FILE NAME BLOCK.
                                    0717           ;
      57    00000000'EF         DE  0717           MOVAL   L^MAC$GL_RESULT, R7     ;SAVE THE RESULT NAME STRING
                     67         90  071E           MOVB    DSC$W_LENGTH(R7),-
            00000003'EF             0720             L^MAC$LISPT_RLFNM+NAM$B_RSL
                                    0725                                           ;COPY LENGTH OF EXPANDED STRING
00000000'EF    04 B7   67       28  0725           MOVC3   DSC$W_LENGTH(R7), @DSC$A_POINTER(R7), -
                                    072E                   L^MAC$LIS_NAM_BUF       ;COPY EXPANDED NAME STRING
                                    072E                                           ;FOR OBJECT AND LISTING FILES
      50    00000000'EF         9E  072E           MOVAB   L^MAC$LIST_FAB,R0       ;GET THE FAB ADDRESS
                     20         11  0735           BRB     OPEN_FILE               ;AND GO HANDLE THE REST
                                    0737
                                    0737   OBJ_FILE:
                              0000  0737           .WORD   0                       ;REGISTER SAVE MASK
                                    0739           ;
                                    0739           ; INSERT THE RESULTANT FILENAME INTO THE RELATED FILE NAME BLOCK.
                                    0739           ;
      57    00000000'EF         DE  0739           MOVAL   L^MAC$GL_RESULT, R7     ;SAVE THE RESULT NAME STRING
                     67         90  0740           MOVB    DSC$W_LENGTH(R7),-
            00000003'EF             0742             L^MAC$OBJPT_RLFNM+NAM$B_RSL
                                    0747                                           ;COPY LENGTH OF EXPANDED STRING
00000000'EF    04 B7   67       28  0747           MOVC3   DSC$W_LENGTH(R7), @DSC$A_POINTER(R7), -
                                    0750                   L^MAC$OBJ_NAM_BUF       ;COPY EXPANDED NAME STRING
                                    0750                                           ;FOR OBJECT AND LISTING FILES
      50    00000000'EF         9E  0750           MOVAB   L^MAC$OBJECT_FAB,R0     ;GET THE FAB ADDRESS
                                    0757
                                    0757   OPEN_FILE:
           51    04 AC          D0  0757           MOVL    4(AP), R1               ;GET THE RESULTING FILENAME
                 04 A1          D0  075B           MOVL    DSC$A_POINTER(R1),-     ;STORE ADDRESS OF FILENAME STRING
                 2C A0              075E                   FAB$L_FNA(R0)           ;       IN THE FAB
                    61          90  0760           MOVB    DSC$W_LENGTH(R1),-      ;STORE LENGTH OF FILENAME STRING
                 34 A0              0762                   FAB$B_FNS(R0)           ;       IN THE FAB
           02 A0    00          B0  0764           MOVW    #0, FAB$W_IFI(R0)       ; CLEAR IFI... VALIDATES THE FAB
                                04  0768           RET
                                    0769
```

```
                          0769  1076          .SBTTL  GET_KEYWORDS Obtain qualifier keywords
                          0769  1077 ;++
                          0769  1078 ;
                          0769  1079 ; Functional Description:
                          0769  1080 ;
                          0769  1081 ;        This routine is called to obtain the next (if any) keyword for the
                          0769  1082 ;        indicated qualifier.
                          0769  1083 ;
                          0769  1084 ; Inputs:
                          0769  1085 ;
                          0769  1086 ;        (SP)    adr     Return address (result of JSB entry)
                          0769  1087 ;        4(SP)   adr     Address of a dynamic string descriptor
                          0769  1088 ;        R7      adr     Address of a dynamic string descriptor for the qualifier
                          0769  1089 ;
                          0769  1090 ; Outputs:
                          0769  1091 ;
                          0769  1092 ;        4(SP)   adr     The descriptor now points to the latest keyword parsed
                          0769  1093 ;
                          0769  1094 ; Routine Value:
                          0769  1095 ;
                          0769  1096 ;        R0      contains the return status from the CLI.
                          0769  1097 ;
                          0769  1098 ;--
                          0769  1099
                          0769  1100 GET_KEYWORDS:
              04 AE   DD  0769  1101         PUSHL   4(SP)                   ; Push address of descriptor.
                 57   DD  076C  1102         PUSHL   R7                      ; Push the address of the qualifier descript
00000000'GF      02   FB  076E  1103         CALLS   #2, G^CLI$GET_VALUE     ; Get the next keyword.
                      05  0775  1104         RSB
                          0776  1105
                          0776  1106
```

```
                                   0776  1108          .SBTTL  PROCESS CROSS QUALIFIER
                                   0776  1109
                                   0776  1110 ;++
                                   0776  1111 ; FUNCTIONAL DESCRIPTION:
                                   0776  1112 ;
                                   0776  1113 ;         THE KEYWORDS ARE SCANNED AND MAC$GL_CRF_FLG IS FILLED IN WITH
                                   0776  1114 ;         THE CORRESPONDING BITS.
                                   0776  1115 ;
                                   0776  1116 ;--
                                   0776  1117
                                   0776  1118 MACR_CROS:
                             0FFC  0776  1119          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTER SAVE MASK
     5B    00000000'EF         9E  0778  1120          MOVAB   L^MAC$GL_FLAGS,R11          ;POINT R11 TO FLAGS
                 04 AC         DD  077F  1121          PUSHL   4(AP)                       ; Address of dynamic string descriptor.
     57    00000016'EF         9E  0782  1122          MOVAB   CROSS, R7                   ; Address of qualifier descriptor.
                 DD AF         16  0789  1123          JSB     GET_KEYWORDS                ; Get the first keyword (if any).
                 0E 50         E8  078C  1124          BLBS    R0, 15$                     ; Got one, go look it up!
                    12         3C  078F  1125          MOVZWL  #CRF$M_SYMBOLS!CRF$M_MACROS,- ;NO--SET THE DEFAULT
           00000000'EF             0791  1126                          L^MAC$GL_CRF_FLG       ;OF SYMBOLS AND MACROS
                               04  0796  1127          RET                                 ;AND LEAVE IT AT THAT
                                   0797  1128
                                   0797  1129 ;
                                   0797  1130 ; LOOP, LOOKING UP KEYWORDS AND SETTING BITS IN MAC$GL_CRF_FLG
                                   0797  1131 ;
                 CF AF         16  0797  1132 10$:     JSB     GET_KEYWORDS                ; Get the keyword
                 4B 50         E9  079A  1133          BLBC    R0, 40$                     ; All done?
                                   079D  1134
            56      6E         D0  079D  1135 15$:     MOVL    (SP), R6                    ; Get the descriptor's address.
            50      66         3C  07A0  1136          MOVZWL  DSC$W_LENGTH(R6), R0        ;GET LENGTH OF ASCII STRING
                    43         13  07A3  1137          BEQL    40$                         ;IF EQL THEN ALL DONE
     51    00000000'EF         9E  07A5  1138          MOVAB   L^MAC$AB_TMPSYM,R1          ;THERE IS A STRING--PT TO NAME BLK
            81      50         90  07AC  1139          MOVB    R0,(R1)+                    ;STORE LENGTH OF STRING
         04 B6      50         2C  07AF  1140          MOVC5   R0,@DSC$A_POINTER(R6),-     ;COPY NAME INTO TMPSYM
       61     1F    00             07B3  1141                  #0,#SYM$K_MAXLEN,(R1)       ;WITH ZERO FILL
     55    00000000'EF         9E  07B6  1142          MOVAB   L^MAC$CRF_OPTIONS,R5        ;POINT TO THE LIST OF OPTIONS
                 F840'         30  07BD  1143          BSBW    MAC$SRC_KEYS                ;LOOK UP THE OPTION
                 0A 50         E8  07C0  1144          BLBS    R0,20$                      ;BRANCH IF FOUND
  00000000'GF       00         FB  07C3  1145          CALLS   #0,G^MAC$ERR_KEY_WD         ;ERROR--REPORT IT
                 F833'         31  07CA  1146          BRW     MAC$LAST_CHANCE             ;AND EXIT
        50     05 A1           D0  07CD  1147 20$:     MOVL    SYM$L_VAL(R1),R0            ;GET THE VALUE
                    0C         14  07D1  1148          BGTR    30$                         ;IF GTR THEN OK
            50      50         D2  07D3  1149          MCOML   R0,R0                       ;ELSE THIS IS "NOXXXX"
  00000000'EF       50         CA  07D6  1150          BICL2   R0,L^MAC$GL_CRF_FLG         ;CLEAR THE CREF FLAG
                    B8         11  07DD  1151          BRB     10$                         ;CONTINUE
  00000000'EF       50         C8  07DF  1152 30$:     BISL2   R0,L^MAC$GL_CRF_FLG         ;SET IN CREF FLAGS
                    AF         11  07E6  1153          BRB     10$                         ;GO GET NEXT OPTION
                               04  07E8  1154 40$:     RET                                 ;RETURN TO CLI
                                   07E9  1155
```

```
                         07E9          .SBTTL  PROCESS SHOW/NOSHOW,ENABLE/DISABLE QUALIFIERS
                         07E9
                         07E9   ;++
                         07E9   ; FUNCTIONAL DESCRIPTION:
                         07E9   ;
                         07E9   ;          THIS ROUTINE IS CALLED TO PROCESS THE /SHOW QUALIFIER.
                         07E9   ;          THE KEYWORDS ARE SCANNED AND THE APPROPRIATE BITS ARE
                         07E9   ;          SET IN THE LIST/ENABLE FLAGS WORD, MAC$GL_ENLISF
                         07E9   ;
                         07E9   ;--
                         07E9
                         07E9   MACR_SHOW:
                 0FFC    07E9          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTER SAVE MASK
         5A    01   D0   07EB          MOVL    #1,R10                     ;FLAG SETTING OPTIONS
57   0000009D'EF    9E   07EE          MOVAB   SHOW, R7                   ; Address of qualifier descriptor.
               0B   11   07F5          BRB     SHOW_NOSH                  ;
                         07F7
                         07F7   MACR_NOSH:
                 0FFC    07F7          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTER SAVE MASK
               5A   D4   07F9          CLRL    R10                        ;FLAG CLEARING OPTIONS
57   00000081'EF    9E   07FB          MOVAB   NOSHOW, R7                 , Address of qualifier descriptor.
                         0802
                         0802   SHOW_NOSH:
59   00000000'EF    9E   0802          MOVAB   L^LST$G_DIRLIST,R9         ;POINT TO SHORT NAME TABLE
58   00000000'EF    9E   0809          MOVAB   L^LST$G_LONGNAMES,R8       ;AND LONG NAME TABLE
               27   11   0810          BRB     SHOW_ENAB
                         0812
                         0812   MACR_ENAB:
                 0FFC    0812          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,P9,R10,R11> ;REGISTER SAVE MASK
         5A    01   D0   0814          MOVL    #1,R10                     ;FLAG SETTING OPTIONS
57   00000049'EF    9E   0817          MOVAB   ENABLE, R7                 ; Address of qualifier descriptor.
               0B   11   081E          BRB     ENAB_DSAB                  ;
                         0820
                         0820   MACR_DISA:
                 0FFC    0820          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTER SAVE MASK
               5A   D4   0822          CLRL    R10                        ;FLAG CLEARING OPTIONS
57   0000003A'EF    9E   0824          MOVAB   DISABLE, R7                ; Address of qualifier descriptor.
                         082B
                         082B   ENAB_DSAB:
59   00000000'EF    9E   082B          MOVAB   L^ENB$G_OPTIONS,R9         ;POINT TO SHORT NAME TABLE
58   00000000'EF    9E   0832          MOVAB   L^ENB$G_LONGNAMES,R8       ;AND LONG NAME TABLE
                         0839
                         0839   ;
                         0839   ; Fall into the common code...
                         0839   ;
                         0839
```

```
                                0839   1204 ;
                                0839   1205 ; COMMON CODE FOR ENABLE/DISABLE/SHOW/NOSHOW.  THE VALUES ARE SCANNED,
                                0839   1206 ; BITS SET OR CLEARED IN SYMBOL BLOCK, AND BITS SET OR CLEARED IN
                                0839   1207 ; MAC$GL_ENLISF (COMMAND ENABLE / LIST FLAGS) UNTIL NO MORE VALUES.
                                0839   1208 ;
                                0839   1209 SHOW_ENAB:
       5B    00000000'EF   9E   0839   1210          MOVAB   L^MAC$GL_FLAGS,R11          ;R11 MUST POINT AT FLAGS
                   04 AC   DD   0840   1211          PUSHL   4(AP)                       ; Address of dynamic string descriptor.
                 FF22 CF   16   0843   1212          JSB     GET_KEYWORDS                ; No, any keywords specified?
                   08 50   E8   0847   1213          BLBS    R0, 15$                     ; Yes, process them.
                           04   084A   1214          RET                                 ;* *SHOULD DO DEFAULT THING!
                                084B   1215
                 FF1A CF   16   084B   1216 10$:     JSB     GET_KEYWORDS                ; Get the next keyword.
                   6C 50   E9   084F   1217          BLBC    R0, 50$                     ; All done, return.
                                0852   1218
             56      6E    D0   0852   1219 15$:     MOVL    (SP), R6                    ; Get the descriptor's address.
             50      66    3C   0855   1220          MOVZWL  DSC$W_LENGTH(R6), R0        ;ANOTHER VALUE?
                     64    13   0858   1221          BEQL    50$                         ;IF EQL NO--EXIT
       51    00000000'EF   9E   085A   1222          MOVAB   L^MAC$AB_TMPSYM,R1          ;COPY TO TEMP BUFFER
             81      50    90   0861   1223          MOVB    R0,(R1)+                    ;COPY NAME LENGTH
          04 B6      50    2C   0864   1224          MOVC5   R0,@DSC$A_POINTER(R6),-     ;COPY NAME INTO TMPSYM
        61   1F      00         0868   1225                  #0,#SYM$K_MAXLEN,(R1)       ;...
             55      59    D0   086B   1226          MOVL    R9,R5                       ;LOOK UP NAME IN SHORT NAME TABLE
                  F78F'    30   086E   1227          BSBW    MAC$SRC_LIST                ;IN THE SHORT NAMES
                   19 50   E8   0871   1228          BLBS    R0,30$                      ;BRANCH IF WE FOUND IT
             55      58    D0   0874   1229          MOVL    R8,R5                       ;NO--TRY THE LONG NAMES
                  F786'    30   0877   1230          BSBW    MAC$SRC_KEYS                ;...
                   0A 50   E8   087A   1231          BLBS    R0,20$                      ;BRANCH IF WE FOUND IT
     00000000'GF     00    FB   087D   1232          CALLS   #0,G^MAC$ERR_KEY_WD         ;NO--ERROR--REPORT PROBLEM
                  F779'    31   0884   1233          BRW     MAC$LAST_CHANCE             ;AND EXIT
          51       05 A1   D0   0887   1234 20$:     MOVL    SYM$L_VAL(R1),R1            ;POINT TO THE REAL OPTION NAME BLOCK
                      BE   13   088B   1235          BEQL    10$                         ;BRANCH IF "NONE" (/SHOW W/NO VALUES)
                   18 5A   E9   088D   1236 30$:     BLBC    R10,40$                     ;BRANCH IF CLEARING OPTIONS
   00000000'EF     09 A1   A8   0890   1237          BISW2   SYM$W_FLAG(R1),L^MAC$GL_ENLISF ;NO--SET IN COMMAND FLAGS
         05 A1     FF 8F   98   0898   1238          CVTBL   #-1,SYM$L_VAL(R1)           ;SET FLAG IN SYMBOL BLOCK
   00000000'EF     09 A1   AA   089D   1239          BICW2   SYM$W_FLAG(R1),L^MAC$GL_DSLISF ;NOTE FLAG SET, IN DISABLE FLAGS
                    FFA3   31   08A5   1240          BRW     10$                         ;PROCESS NEXT OPTION
   00000000'EF     09 A1   AA   08A8   1241 40$:     BICW2   SYM$W_FLAG(R1),L^MAC$GL_ENLISF ;CLEAR IN COMMAND FLAGS
                   05 A1   D4   08B0   1242          CLRL    SYM$L_VAL(R1)               ;AND IN SYMBOL BLOCK
   00000000'EF     09 A1   A8   08B3   1243          BISW2   SYM$W_FLAG(R1),L^MAC$GL_DSLISF ;NOTE FLAG CLEARED, IN DISABLE FLAGS
                    FF8D   31   08BB   1244          BRW     10$                         ;PROCESS NEXT OPTION
                           04   08BE   1245 50$:     RET
                                08BF   1246
```

```
                        08BF          .SBTTL  PROCESS DEBUG/NODEBUG QUALIFIERS
                        08BF  ;++
                        08BF  ; FUNCTIONAL DESCRIPTION:
                        08BF  ;
                        08BF  ;       The /DEBUG qualifier (plus its associated parameters) issued with the
                        08BF  ;       DCL MACRO command will behave in the same manner as issuing a com-
                        08BF  ;       bination of the /ENABLE and /DISABLE qualifiers utilizing the DEBUG
                        08BF  ;       and TRACEBACK parameters.
                        08BF  ;
                        08BF  ;       This routine searches for a specific parameter for the /DEBUG command.
                        08BF  ;       Upon recognition of a parameter, the associated action (setting and/or
                        08BF  ;       clearing bits) is performed.  The list of parameters and thier
                        08BF  ;       value/actions are enumerated below.
                        08BF  ;--
                        08BF
                        08BF  MACR_NODE:
                  0FFC  08BF          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTER SAVE MASK
51   00000000'EF    9E  08C1          MOVAB   DBG$G_LONGNAMES,R1      ;DISABLE BOTH DEBUG AND TRACEBACK
            0052    31  08C8          BRW     DEBU_NODE               ;NO NEED TO DO ANY LOOK UPS, JUST DO IT
                        08CB
                        08CB  MACR_DEBU:
                  0FFC  08CB          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;REGISTER SAVE MASK
              5A    D4  08CD          CLRL    R10                     ;FLAG FOR SYMBOLS AND/OR TRACEBACK SEEN
5B   00000000'GF    9E  08CF          MOVAB   G^MAC$GL_FLAGS, R11     ;POINT TO FLAGS.
           04 AC    DD  08D6          PUSHL   4(AP)                   ; Address of dynamic string descriptor.
57   0000002D'EF    9E  08D9          MOVAB   DEBUG, R7               ; Push address of DEBUG qualifier descriptor
                        08E0
                        08E0  DBG_NEXT_OPT:
         FE85 CF    16  08E0          JSB     GET_KEYWORDS            ; Get the keywords (if any).
           03 50    E8  08F4          BLBS    R0, 5$                  ; Got one, process it!
            0109    31  08E7          BRW     DBG_RETURN              ; No more, so return (Will handle default ca
                        08EA
        56    6E    D0  08EA  5$:     MOVL    (SP), R6                ; Get the descriptor's address.
        50    66    3C  08ED          MOVZWL  DSC$W_LENGTH(R6),R0     ;ANOTHER VALUE?
              03    12  08F0          BNEQ    10$                     ;IF NEQ YES -- CONTINUE PROCESSING IT
            00FE    31  08F2          BRW     DBG_RETURN              ;IF EQL NO -- EXIT
                        08F5
51   00000000'EF    9E  08F5  10$:    MOVAB   L^MAC$AB_TMPSYM,R1      ;COPY TO TEMP BUFFER
        81    50    90  08FC          MOVB    R0,(R1)+                ;COPY NAME LENGTH
     04 B6    50    2C  08FF          MOVC5   R0,@DSC$A_POINTER(R6),- ;COPY NAME INTO TMPSYM
    61  1F    00        0903                  #0,#SYM$K_MAXLEN,(R1)   ;...
55   00000000'GF    9E  0906          MOVAB   G^DBG$G_LONGNAMES,R5    ;LOOK UP THE OPTION IN THE LONG NAMES
           F6F0'    30  090D          BSBW    MAC$SRC_KEYS            ;...
           0A 50    E8  0910          BLBS    R0,DEBU_NODE            ;BRANCH IF WE FOUND IT
00000000'GF   00    FB  0913          CALLS   #0,G^MAC$ERR_KEY_WD     ;NO--ERROR--REPORT PROBLEM
           F6E3'    31  091A          BRW     MAC$LAST_CHANCE         ;AND EXIT
                        091D
```

```
                                  091D  1296 ;++
                                  091D  1297 ; COMMON CODE FOR DEBUG/NODEBUG.
                                  091D  1298 ; Bits are set/cleared in the symbol block and in MAC$GL_ENLISF
                                  091D  1299 ; (Command Enable/List Flags) until no more values are encountered.
                                  091D  1300 ;
                                  091D  1301 ; First execute some common code to set up addressing of the necessary
                                  091D  1302 ; symbol blocks.
                                  091D  1303 ;--
                                  091D  1304
                                  091D  1305 DEBU_NODE:
      52  00000000'GF     9E      091D  1306          MOVAB   G^ENB$G_DEBUG,R2         ;POINT TO DEBUG SYMBOL BLOCK ENTRY
      53  00000000'GF     9E      0924  1307          MOVAB   G^ENB$G_TRACEBACK,R3     ;AND TO THE TRACEBACK ENTRY.
                                  092B  1308
                                  092B  1309 ;++
                                  092B  1310 ; Begin looking for the specific qualifier parameters...
                                  092B  1311 ;
                                  092B  1312 ; The parameter code values are...
                                  092B  1313 ;
                                  092B  1314 ;       Value      Qualifier         Action
                                  092B  1315 ;         0        None              Disable both the DEBUG and TRACEBACK options
                                  092B  1316 ;         1        Symbols           Disable TRACEBACK and Enable DEBUG option
                                  092B  1317 ;         2        Traceback         Disable DEBUG and Enable TRACEBACK option
                                  092B  1318 ;         3        All (default)     Enable both the DEBUG and TRACEBACK options
                                  092B  1319 ;--
                                  092B  1320
           51      05 A1    D0    092B  1321          MOVL    SYM$L_VAL(R1),R1          ;CHECK WHICH OPTION WAS SELECTED
                      29    14    092F  1322          BGTR    10$                       ;BRANCH IF PARAM IS NOT "NONE"
   00000000'GF     09 A2    AA    0931  1323          BICW2   SYM$W_FLAG(R2),G^MAC$GL_ENLISF   ;SINCE IT IS NONE, WE DISABLE
                   05 A2    D4    0939  1324          CLRL    SYM$L_VAL(R2)             ;BOTH THE DEBUG AND THE TRACE-
   00000000'GF     09 A3    AA    093C  1325          BICW2   SYM$W_FLAG(R3),G^MAC$GL_ENLISF   ;BACK QUALIFIER BITS IN THE
                   05 A3    D4    0944  1326          CLRL    SYM$L_VAL(R3)             ;GLOBAL AND LOCAL CONTROL BLOCKS.
   00000000'EF     09 A2    A8    0947  1327          BISW2   SYM$W_FLAG(R2),L^MAC$GL_DSLISF ;NOTE FLAG CLEARED, IN DISABLE FLAGS
   00000000'EF     09 A3    A8    094F  1328          BISW2   SYM$W_FLAG(R3),L^MAC$GL_DSLISF ;NOTE FLAG CLEARED, IN DISABLE FLAGS
                    0099    31    0957  1329          BRW     DBG_RETURN                ;ALL DONE, CONTINUE OTHER PROCESSING.
                                  095A  1330
              01       51   D1    095A  1331 10$:     CMPL    R1,#1                     ;CHECK FOR THE "SYMBOLS" QUAL
                       31   14    095D  1332          BGTR    20$                       ;BRANCH IF NOT "SYMBOLS"
   00000000'EF     09 A2    A8    095F  1333          BISW2   SYM$W_FLAG(R2),L^MAC$GL_ENLISF   ;SET IN COMMAND FLAGS
         05 A2     FF 8F    98    0967  1334          CVTBL   #-1,SYM$L_VAL(R2)         ;SET FLAG IN SYMBOL BLOCK
   00000000'EF     09 A2    AA    096C  1335          BICW2   SYM$W_FLAG(R2),L^MAC$GL_DSLISF ;NOTE FLAG SET, IN DISABLE FLAGS
                   16 5A    E8    0974  1336          BLBS    R10,15$                   ;TRACEBACK or SYMBOLS PREVIOUSLY SET?
   00000000'GF     09 A3    AA    0977  1337          BICW2   SYM$W_FLAG(R3),G^MAC$GL_ENLISF   ;DISABLE TRACEBACK IN COMMAND
                   05 A3    D4    097F  1338          CLRL    SYM$L_VAL(R3)             ;FLAGS AND SYMBOL BLOCK
   00000000'EF     09 A3    A8    0982  1339          BISW2   SYM$W_FLAG(R3),L^MAC$GL_DSLISF ;NOTE FLAG CLEARED, IN DISABLE FLAGS
              5A       01   D0    098A  1340          MOVL    #1,R10                    ;SHOW THAT "SYMBOLS" HAS BEEN SEEN.
                    FF50    31    098D  1341 15$:     BRW     DBG_NEXT_OPT              ;PROCESS NEXT OPTION
                                  0990  1342
              02       51   D1    0990  1343 20$:     CMPL    R1,#2                     ;CHECK FOR THE "TRACEBACK" QUAL
                       31   14    0993  1344          BGTR    30$                       ;BRANCH IF NOT "TRACEBACK"
   00000000'GF     09 A3    A8    0995  1345          BISW2   SYM$W_FLAG(R3),G^MAC$GL_ENLISF   ;SET IN COMMAND FLAGS
         05 A3     FF 8F    98    099D  1346          CVTBL   #-1,SYM$L_VAL(R3)         ;SET FLAG IN SYMBOL BLOCK
   00000000'EF     09 A3    AA    09A2  1347          BICW2   SYM$W_FLAG(R3),L^MAC$GL_DSLISF ;NOTE FLAG SET, IN DISABLE FLAGS
                   16 5A    E8    09AA  1348          BLBS    R10,25$                   ;SYMBOLS or TRACEBACK PREVIOUSLY SET?
   00000000'EF     09 A2    AA    09AD  1349          BICW2   SYM$W_FLAG(R2),L^MAC$GL_ENLISF   ;DISABLE DEBUG IN COMMAND
                   05 A2    D4    09B5  1350          CLRL    SYM$L_VAL(R2)             ;FLAGS AND SYMBOL BLOCK
   00000000'EF     09 A2    A8    09B8  1351          BISW2   SYM$W_FLAG(R2),L^MAC$GL_DSLISF ;NOTE FLAG CLEARED, IN DISABLE FLAGS
              5A       01   D0    09C0  1352          MOVL    #1,R10                    ;SHOW THAT "TRACEBACK" HAS BEEN SEEN.
```

```
                    FF1A   31  09C3  25$:    BRW     DBG_NEXT_OPT            ;PROCESS NEXT OPTION
                               09C6
                               09C6  ;++
                               09C6  ; If you are here, we assume the ALL parameter was specified...
                               09C6  ; In other words Both the DEBUG and TRACEBACK options are enabled.
                               09C6  ;--
                               09C6
      00000000'GF   09 A2  A8  09C6  30$:    BISW2   SYM$W_FLAG(R2),G^MAC$GL_ENLISF
            05 A2   FF 8F  98  09CE          CVTBL   #-1,SYM$L_VAL(R2)
      00000000'GF   09 A3  A8  09D3          BISW2   SYM$W_FLAG(R3),G^MAC$GL_ENLISF
            05 A3   FF 8F  98  09DB          CVTBL   #-1,SYM$L_VAL(R3)
      00000000'EF   09 A2  AA  09E0          BICW2   SYM$W_FLAG(R2),L^MAC$GL_DSLISF ;NOTE FLAG SET, IN DISABLE FLAGS
      00000000'EF   09 A3  AA  09E8          BICW2   SYM$W_FLAG(R3),L^MAC$GL_DSLISF ;NOTE FLAG SET, IN DISABLE FLAGS
                    FEED   31  09F0          BRW     DBG_NEXT_OPT
                               09F3
                               09F3  ;++
                               09F3  ; That's all for now...continue with the other processing
                               09F3  ;--
                               09F3
                           04  09F3  DBG_RETURN:     RET
```

```
                                  09F4           .SBTTL  PROCESS /UPDATE QUALIFIER
                                  09F4
                                  09F4   ;++
                                  09F4   ; Functional description
                                  09F4   ;
                                  09F4   ;       This routine is called to process the /UPDATE qualifer.
                                  09F4   ;       The Source update merge procedure SUM$UPDATE_QUAL is called to
                                  09F4   ;       process any values on the qualifier
                                  09F4   ;
                                  09F4   ; Inputs:
                                  09F4   ;
                                  09F4   ;       4(AP)   adr     Address of a dynamic string descriptor.
                                  09F4   ;
                                  09F4   ; Implicit Inputs:
                                  09F4   ;
                                  09F4   ;       CLI$GET_VALUE   adr     Address of CLI call back routine.
                                  09F4   ;       UPDATE          adr     Address of the qualifier name descriptor.
                                  09F4   ;       UPDATE_LIST     adr     Address of list head for the update file list.
                                  09F4   ;
                                  09F4   ; Outputs:
                                  09F4   ;
                                  09F4   ;       Update files list
                                  09F4   ;
                                  09F4   ;--
                                  09F4
                                  09F4   MACR_UPDA:
                            0000  09F4           .WORD   0                       ; Register save mask
                    04 AC     DD  09F6           PUSHL   4(AP)                   ; Address of the descriptor.
              00000000'EF     DF  09F9           PUSHAL  UPDATE_LIST             ; Address to put update files list
              000000A9'EF     9F  09FF           PUSHAB  UPDATE                  ; Address of the qualifier descriptor.
              00000000'GF     9F  0A05           PUSHAB  G^CLI$GET_VALUE         ; Address of the CLI call back routine.
      00000000'GF    04       FB  0A0B           CALLS   #4,G^SUM$UPDATE_QUAL    ; Process list
                              04  0A12           RET                             ; Return to finish input file processing
                                  0A13
                                  0A13           .END
```

```
$$.TMP1                          = 00000002                  FLG$M_ALLCHR                    = 00000001
$$.TMP2                          = 00000066                  FLG$M_BOL                       = 00000002
BIT...                           = 00000007                  FLG$M_CHKLPND                   = 00100000
CLI$GET_VALUE                      ********    X   05        FLG$M_COMPEXPR                  = 00000004
CLI$PRESENT                        ********    X   05        FLG$M_CONT                      = 00000008
CLI$_COMMA                         ********    X   05        FLG$M_CRF                       = 40000000
CLI$_CONCAT                        ********    X   05        FLG$M_CRSEEN                    = 00000001
CLI$_LOCNEG                        ********    X   05        FLG$M_DATRPT                    = 00000010
CLI$_LOCPRES                       ********    X   05        FLG$M_DBGOUT                    = 00004000
CLI$_NEGATED                       ********    X   05        FLG$M_DLIMSTR                   = 00008000
COMMAND_LINE                       00000009 R      04        FLG$M_ENDMCH                    = 00000020
CRF$M_DEFAULT                    = 00000012                  FLG$M_EVALEXPR                  = 00000040
CRF$M_DIR                        = 00000001                  FLG$M_EXPOPT                    = 00000080
CRF$M_MACROS                     = 00000002                  FLG$M_EXTERR                    = 00010000
CRF$M_OPCODES                    = 00000004                  FLG$M_EXTWRN                    = 00020000
CRF$M_REGISTERS                  = 00000008                  FLG$M_FIRSTLN                   = 00000200
CRF$M_SYMBOLS                    = 00000010                  FLG$M_IFSTAT                    = 00800000
CRF$V_DIR                        = 00000000                  FLG$M_IIF                       = 00400000
CRF$V_MACROS                     = 00000001                  FLG$M_INSERT                    = 00000100
CRF$V_OPCODES                    = 00000002                  FLG$M_IRPC                      = 20000000
CRF$V_REGISTERS                  = 00000003                  FLG$M_LEXOP                     = 00000002
CRF$V_SYMBOLS                    = 00000004                  FLG$M_LSTXST                    = 00000200
CROSS                              00000016 R      04        FLG$M_MAC2COL                   = 00000800
DBG$G_LONGNAMES                    ********    X   05        FLG$M_MACL                      = 00000800
DBG_NEXT_OPT                       000008E0 R      05        FLG$M_MACLTB                    = 08000000
DBG_RETURN                         000009F3 R      05        FLG$M_MACTXT                    = 00010000
DEBUG                              0000002D R      04        FLG$M_MEBLST                    = 00001000
DEBU_NODE                          0000091D R      05        FLG$M_MOREARG                   = 00002000
DISABLE                            0000003A R      04        FLG$M_MOREINP                   = 00000008
DSC$A_POINTER                    = 00000004                  FLG$M_NEWPND                    = 00000400
DSC$B_CLASS                      = 00000003                  FLG$M_NOREF                     = 01000000
DSC$B_DTYPE                      = 00000002                  FLG$M_NTYPEPC                   = 00000020
DSC$K_CLASS_D                    = 00000002                  FLG$M_NULCHR                    = 00040000
DSC$K_DTYPE_T                    = 0000000E                  FLG$M_OBJXST                    = 00200000
DSC$W_LENGTH                     = 00000000                  FLG$M_OPNDCHK                   = 00000100
ENABLE                             00000049 R      04        FLG$M_OPRND                     = 00002000
ENAB_DSAB                          0000082B R      05        FLG$M_OPTVFLIDX                 = 00001000
ENB$G_DEBUG                        ********    X   05        FLG$M_ORDLST                    = 00020000
ENB$G_LONGNAMES                    ********    X   05        FLG$M_P2                        = 00004000
ENB$G_OPTIONS                      ********    X   05        FLG$M_RPTIRP                    = 10000000
ENB$G_TRACEBACK                    ********    X   05        FLG$M_SEQFIL                    = 02000000
FAB$B_BID                        = 00000000                  FLG$M_SKAN                      = 00008000
FAB$B_DNS                        = 00000035                  FLG$M_SPECOP                    = 00000004
FAB$B_FAC                        = 00000016                  FLG$M_SPLALL                    = 04000000
FAB$B_FNS                        = 00000034                  FLG$M_STOIMF                    = 00040000
FAB$C_BID                        = 00000003                  FLG$M_SYM2COL                   = 00000400
FAB$C_BLN                        = 00000050                  FLG$M_TOCFLG                    = 00080000
FAB$L_DNA                        = 00000030                  FLG$M_UPAFLG                    = 00000010
FAB$L_FNA                        = 0000002C                  FLG$M_UPDFIL                    = 00000080
FAB$L_FOP                        = 00000004                  FLG$M_UPMARG                    = 00000040
FAB$L_NAM                        = 00000028                  FLG$M_XCRF                      = 80000000
FAB$L_XAB                        = 00000024                  FLG$V_ALLCHR                    = 00000000
FAB$M_GET                        = 00000002                  FLG$V_BOL                       = 00000001
FAB$M_NAM                        = 01000000                  FLG$V_CHKLPND                   = 00000014
FAB$V_OFP                        = 0000001D                  FLG$V_COMPEXPR                  = 00000002
FAB$W_IFI                        = 00000002                  FLG$V_CONT                      = 00000003
FILE                               00000057 R      04        FLG$V_CRF                       = 0000001E
```

```
FLG$V_CRSEEN                 = 00000020               LIB$GET_VM                       ********   X    05
FLG$V_DATRPT                 = 00000004               LIBRARY                          00000063 R      04
FLG$V_DBGOUT                 = 0000002E               LIBR_FILE                        0000063B R      05
FLG$V_DLIMSTR                = 0000002F               LISTING                          00000072 R      04
FLG$V_ENDMCH                 = 00000005               LIST_FILE                        00000715 R      05
FLG$V_EVALEXPR               = 00000006               LIST_QUALS                       00000304 RG     05
FLG$V_EXPOPT                 = 00000007               LOC_LST_FLAG                     00000004 R      03
FLG$V_EXTERR                 = 00000030               LOC_OBJ_FLAG                     00000008 R      03
FLG$V_EXTWRN                 = 00000031               LST$G_DIRLIST                    ********   X    05
FLG$V_FIRSTLN                = 00000029               LST$G_LONGNAMES                  ********   X    05
FLG$V_IFSTAT                 = 00000017               MAC$AB_TMPSYM                    ********   X    05
FLG$V_IIF                    = 00000016               MAC$ALL_MLB_MLF                  ********   X    05
FLG$V_INSERT                 = 00000008               MAC$CRF_OPTIONS                  ********   X    05
FLG$V_IRPC                   = 0000001D               MAC$ERR_FIND_INP                 ********   X    05
FLG$V_LEXOP                  = 00000021               MAC$ERR_KEY_QD                   ********   X    05
FLG$V_LSTXST                 = 00000009               MAC$ERR_LBROPEN                  ********   X    05
FLG$V_MAC2COL                = 0000002B               MAC$ERR_NOMEM_O                  ********   X    05
FLG$V_MACL                   = 0000000B               MAC$ERR_OPN_INP                  ********   X    05
FLG$V_MACLTB                 = 0000001B               MAC$ERR_OPN_OUT                  ********   X    05
FLG$V_MACTXT                 = 00000010               MAC$ERR_TEXT                     ********   X    05
FLG$V_MEBLST                 = 0000000C               MAC$GC_FFLAGS                    ********   X    05
FLG$V_MOREARG                = 0000002D               MAC$GETCMD                       00000000 RG     05
FLG$V_MOREINP                = 00000023               MAC$GL_BASEADDR                  ********   X    05
FLG$V_NEWPND                 = 0000000A               MAC$GL_CMDLEN                    ********   X    05
FLG$V_NOREF                  = 00000018               MAC$GL_CMDLIN                    ********   X    05
FLG$V_NTYPEPC                = 00000025               MAC$GL_CONTEXT                   ********   X    05
FLG$V_NULCHR                 = 00000032               MAC$GL_CRF_FLG                   ********   X    05
FLG$V_OBJXST                 = 00000015               MAC$GL_CTLMSK                    ********   X    05
FLG$V_OPNDCHK                = 00000028               MAC$GL_DIRFLG                    ********   X    05
FLG$V_OPRND                  = 0000000D               MAC$GL_DSLISF                    ********   X    05
FLG$V_OPTVFLIDX              = 0000002C               MAC$GL_ENLISF                    ********   X    05
FLG$V_ORDLST                 = 00000011               MAC$GL_FINPTF                    ********   X    05
FLG$V_P2                     = 0000000E               MAC$GL_FLAGS                     ********   X    05
FLG$V_RPTIRP                 = 0000001C               MAC$GL_INPQUE                    ********   X    05
FLG$V_SEQFIL                 = 00000019               MAC$GL_LIBFUNC                   ********   X    05
FLG$V_SKAN                   = 0000000F               MAC$GL_LIBTYPE                   ********   X    05
FLG$V_SPECOP                 = 00000022               MAC$GL_MLB_CNT                   ********   X    05
FLG$V_SPLALL                 = 0000001A               MAC$GL_MLB_QUE                   ********   X    05
FLG$V_STOIMF                 = 00000012               MAC$GL_RESADDR                   ********   X    05
FLG$V_SYM2COL                = 0000002A               MAC$GL_RESULT                    ********   X    05
FLG$V_TOCFLG                 = 00000013               MAC$GL_STV_ADDR                  ********   X    05
FLG$V_UPAFLG                 = 00000024               MAC$GQ_RNT_CMD                   ********   X    05
FLG$V_UPDFIL                 = 00000027               MAC$G[illegible]AGE              ********   X    05
FLG$V_UPMARG                 = 00000026               MAC$INPPT_RLFNM                  ********   X    05
FLG$V_XCRF                   = 0000001F               MAC$INPUT_XAB                    ********   X    05
GET_INPUT                      0000035E RG     05     MAC$INP_DEFNAM                   ********   X    05
GET_KEYWORDS                   00000769 R      05     MAC$INP_NAM_BUF                  ********   X    05
GET_LISTING                    00000161 RG     05     MAC$LAST_CHANCE                  ********   X    05
GET_LOC_LIST                   00000263 RG     05     MAC$LISPT_RLFNM                  ********   X    05
GET_LOC_OBJECT                 000001D5 RG     05     MAC$LIST_FAB                     ********   X    05
GET_OBJECT                     00000100 RG     05     MAC$LIST_RAB                     ********   X    05
INP_DEFEXT                     00000004 R      04     MAC$LIS_NAM_BUF                  ********   X    05
INP_FILE                       000004BC RG     05     MAC$MLB_DEFNAM                   ********   X    05
INP_LIBR_FILE                  00000645 R      05     MAC$MSG_NO_FILE                  ********   X    05
LBR$INI_CONTROL                ********   X    05     MAC$OBJECT_FAB                   ********   X    05
LBR$OPEN                       ********   X    05     MAC$OBJECT_RAB                   ********   X    05
LIB$FIND_FILE                  ********   X    05     MAC$OBJPT_RLFNM                  ********   X    05
```

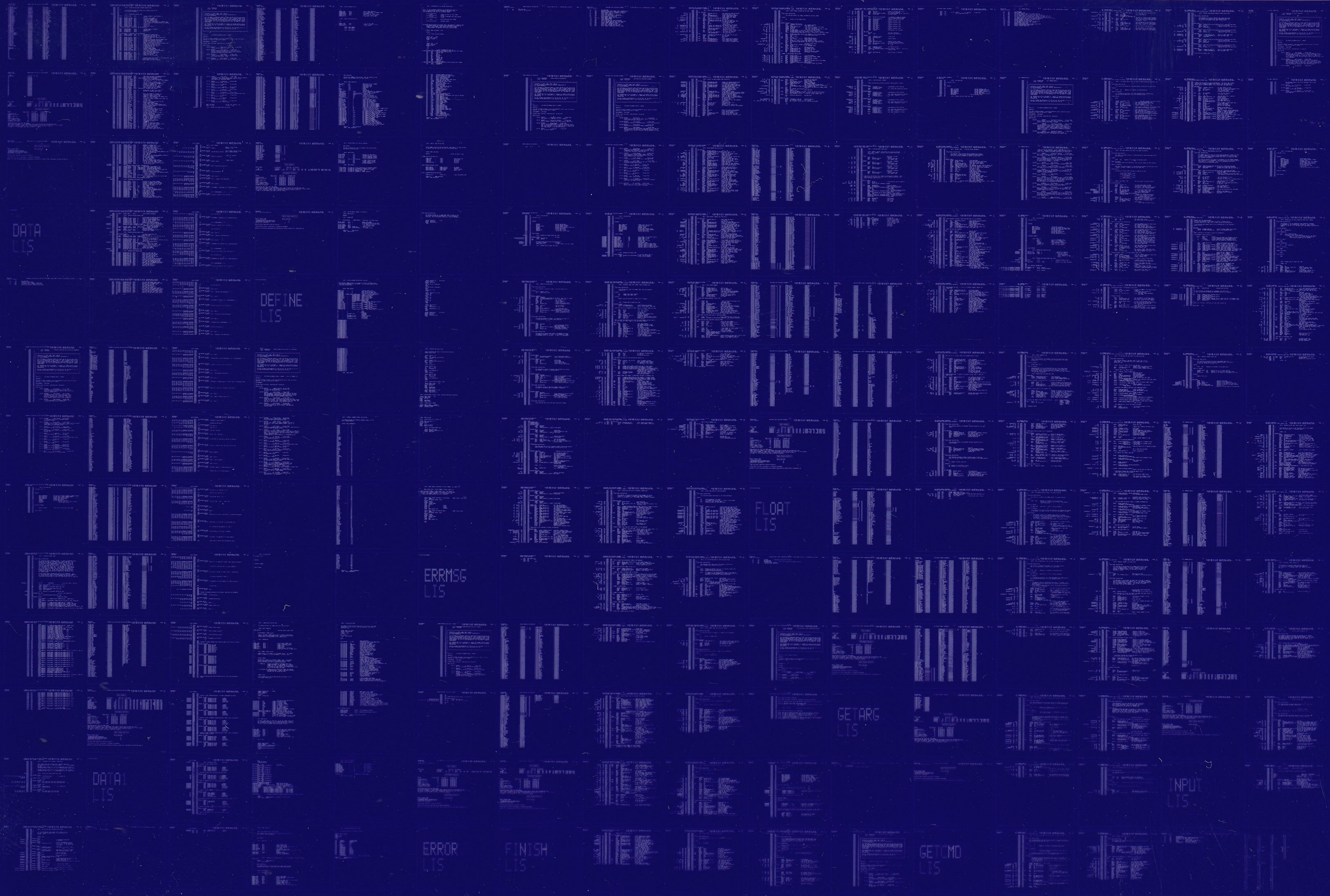
```
MAC$OBJ_NAM_BUF              ********   X   05     PSC$M_ABS                  = FFFFFFF7
MAC$SRC_KEYS                 ********   X   05     PSC$M_ALIGNFLG             = 00004000
MAC$SRC_LIST                 ********   X   05     PSC$M_ALLOPTNS             = 000003FF
MAC$TIMER_OFF                ********   X   05     PSC$M_BYTE                 = 00004000
MAC$TIMER_ON                 ********   X   05     PSC$M_CON                  = FFFFFFFB
MACR_CROS                    00000776 R     05     PSC$M_DEFAULT              = 000001C8
MACR_DEBU                    000008CB R     05     PSC$M_EXE                  = 000000C0
MACR_DISA                    00000820 R     05     PSC$M_GBL                  = 00000010
MACR_ENAB                    00000812 R     05     PSC$M_LCL                  = FFFFFFEF
MACR_NODE                    000008BF R     05     PSC$M_LIB                  = 00000002
MACR_NOSH                    000007F7 R     05     PSC$M_LONG                 = 00004800
MACR_SHOW                    000007E9 R     05     PSC$M_NOEXE                = FFFFFFBF
MACR_UPDA                    000009F4 R     05     PSC$M_NOPIC                = FFFFFFFE
MAC_M_LIBRARY              = 00000020              PSC$M_NORD                 = FFFFFF7F
MAC_M_UPDATE               = 00000040              PSC$M_NOSHR                = FFFFFFDF
MAC_V_CROSSREF             = 00000004              PSC$M_NOVEC                = FFFFFDFF
MAC_V_INPFIL               = 00000003              PSC$M_NOWRT                = FFFFFEFF
MAC_V_LIBRARY              = 00000005              PSC$M_OVR                  = 00000004
MAC_V_LIST                 = 00000001              PSC$M_PAGE                 = 00006400
MAC_V_OBJECT               = 00000002              PSC$M_PIC                  = 00000001
MAC_V_UPDATE               = 00000006              PSC$M_QUAD                 = 00004C00
MLF$K_BLKSIZ                 00000177              PSC$M_RD                   = 00000080
MLF$K_RSFNLN               = 000000FF              PSC$M_REL                  = 00000008
MLF$L_CTINDEX                00000014              PSC$M_SHR                  = 00000020
MLF$L_MCDEF                  00000008              PSC$M_USR                  = FFFFFFFD
MLF$L_QLINK                  0000C000              PSC$M_VEC                  = 00000200
MLF$Q_FNAMDS                 0000000C              PSC$M_WORD                 = 00004400
MLF$T_FNAM                   00000078              PSC$M_WRT                  = 00000180
MLF$X_NAMBLK                 00000018              PSC$S_ALIGNMENT            = 00000004
NAM$B_BID                  = 00000000              PSC$V_ALIGNFLG             = 0000000E
NAM$B_ESL                  = 0000000B              PSC$V_ALIGNMENT            = 0000000A
NAM$B_ESS                  = 0000000A              PSC$V_EXE                  = 00000006
NAM$B_RSL                  = 00000003              PSC$V_GBL                  = 00000004
NAM$B_RSS                  = 00000002              PSC$V_LIB                  = 00000001
NAM$C_BID                  = 00000002              PSC$V_OVR                  = 00000002
NAM$C_BLN                  = 00000060              PSC$V_PIC                  = 00000000
NAM$C_MAXRSS               = 000000FF              PSC$V_RD                   = 00000007
NAM$L_ESA                  = 0000000C              PSC$V_REL                  = 00000003
NAM$L_RLF                  = 00000010              PSC$V_SHR                  = 00000005
NAM$L_RSA                  = 00000004              PSC$V_VEC                  = 00000009
NOSHOW                       00000081 R     04     PSC$V_WRT                  = 00000008
OBJECT                       0000008F R     04     PSC$W_FLAG                   00000009
OBJ_FILE                     00000737 R     05     PSC$W_OPTIONS                0000000D
OPEN_FILE                    00000757 R     05     RESULT                       00000000 R     04
OPEN_OUTPUTS                 00000069 R     05     SHOW                         0000009D R     04
OPF$M_LASTOPR              = 00002000              SHOW_ENAB                    00000839 R     05
OPF$M_OPTEXP               = 00001000              SHOW_NOSH                    00000802 R     05
OPF$V_LASTOPR              = 0000000D              SIZ...                     = 00000001
OPF$V_OPTEXP               = 0000000C              SUM$OPEN                     ********   X   05
PSC$B_NAME                   00000004              SUM$UPDATE_QUAL              ********   X   05
PSC$B_SEG                    0000000C              SYM$B_NAME                   00000004
PSC$B_UNUSED                 0000000B              SYM$B_SEG                    0000000C
PSC$K_BLKSIZ                 00000013              SYM$B_TOKEN                  0000000B
PSC$K_NO_OPTNS             = 0000000A              SYM$K_BLKSIZ                 0000000D
PSC$L_CURLOC                 0000000F              SYM$K_MAXLEN               = 0000001F
PSC$L_LINK                   00000000              SYM$L_LINK                   00000000
PSC$L_MAXLGTH                00000005              SYM$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$W_FLAG                           00000009
SYS$CONNECT                          ********  GX    05
SYS$CREATE                           ********  GX    05
SYS$PARSE                            ********  GX    05
UPDATE                               000000A9 R      04
UPDATE_LIST                          00000000 R      03
X1                                 = 00000400
X2                                 = 0000000F

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

PSECT name                         Allocation          PSECT No.  Attributes
----------                         ----------          ---------  ----------
.  ABS  .                          00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
. BLANK .                          00000000  (     0.)  01 (  1.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$ABS$                              00000177  (   375.)  02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
MAC$RW_DATA                        0000000C  (    12.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD     WRT NOVEC BYTE
MAC$RO_DATA                        000000B7  (   183.)  04 (  4.)  NOPIC   USR   CON   REL   GBL NOSHR NOEXE   RD   NOWRT NOVEC LONG
MAC$RO_CODE_P35                    00000A13  (  2579.)  05 (  5.)  NOPIC   USR   CON   REL   GBL NOSHR   EXE   RD   NOWRT NOVEC LONG
```

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

Phase                        Page faults    CPU Time        Elapsed Time
-----                        -----------    --------        ------------
Initialization                       36    00:00:00.04     00:00:00.60
Command processing                  138    00:00:00.54     00:00:05.40
Pass 1                              336    00:00:07.40     00:00:29.33
Symbol table sort                     0    00:00:00.99     00:00:05.05
Pass 2                              262    00:00:02.47     00:00:10.52
Symbol table output                  46    00:00:00.20     00:00:00.21
Psect synopsis output                 2    00:00:00.03     00:00:00.63
Cross-reference output                0    00:00:00.00     00:00:00.00
Assembler run totals                822    00:00:11.67     00:00:51.74

The working set limit was 1950 pages.
69738 bytes (137 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1005 non-local and 62 local symbols.
1408 source lines were read in Pass 1, producing 50 object records in Pass 2.
18 pages of virtual memory were used to define 16 macros.

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

Macro library name                                Macros defined
------------------                                --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                           0
_$255$DUA28:[MACRO.OBJ]MACRO.MLB;1                       4
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       12
TOTALS (all libraries)                                  16

1004 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

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

0225 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



DATA
LIS

DEFINE
LIS

FLOAT
LIS

ERRMSG
LIS

GETARG
LIS

DATA1
LIS

INPUT
LIS

ERROR
LIS

FINISH
LIS

GETCMD
LIS