```
MMM          MMM     AAAAAAAAA         CCCCCCCCCCC   RRRRRRRRRRR        OOOOOOOOO
MMM          MMM     AAAAAAAAA         CCCCCCCCCCC   RRRRRRRRRRR        OOOOOOOOO
MMM          MMM     AAAAAAAAA         CCCCCCCCCCC   RRRRRRRRRRR        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            RRRRRRRRRRR       OOO         OOO
MMM         MMM    AAA         AAA   CCC            RRRRRRRRRRR       OOO         OOO
MMM         MMM    AAA         AAA   CCC            RRRRRRRRRRR       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      CCCCCCCCCCC   RRR         RRR      OOOOOOOOO
MMM         MMM    AAA         AAA      CCCCCCCCCCC   RRR         RRR      OOOOOOOOO
MMM         MMM    AAA         AAA      CCCCCCCCCCC   RRR         RRR      OOOOOOOOO
```

FILEID**MAIN

```
MM      MM    AAAAAA      IIIIII    NN      NN
MM      MM    AAAAAA      IIIIII    NN      NN
MMMM  MMMM   AA    AA      II      NN      NN
MMMM  MMMM   AA    AA      II      NN      NN
MM  MM  MM   AA    AA      II      NNNN    NN
MM  MM  MM   AA    AA      II      NNNN    NN
MM      MM   AA    AA      II      NN  NN  NN
MM      MM   AA    AA      II      NN  NN  NN
MM      MM   AAAAAAAAAA      II      NN    NNNN
MM      MM   AAAAAAAAAA      II      NN    NNNN
MM      MM   AA    AA      II      NN      NN      ....
MM      MM   AA    AA      II      NN      NN      ....
MM      MM   AA    AA    IIIIII    NN      NN      ....
MM      MM   AA    AA    IIIIII    NN      NN      ....

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     1          .TITLE  MAC$MAIN ENTRY POINT TO VAX-11 MACRO
0000     2          .IDENT  'V04-000'
0000     3
0000     4  ;
0000     5  ;**********************************************************************************
0000     6  ;*                                                                                *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                       *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                        *
0000     9  ;*  ALL RIGHTS RESERVED.                                                          *
0000    10  ;*                                                                                *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED         *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE        *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER        *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY        *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY        *
0000    16  ;*  TRANSFERRED.                                                                  *
0000    17  ;*                                                                                *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE         *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT        *
0000    20  ;*  CORPORATION.                                                                  *
0000    21  ;*                                                                                *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS        *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                       *
0000    24  ;*                                                                                *
0000    25  ;*                                                                                *
0000    26  ;**********************************************************************************
0000    27  ;
0000    28
0000    29  ;++
0000    30  ; FACILITY:     VAX MACRO ASSEMBLER OBJECT LIBRARY
0000    31  ;
0000    32  ; ABSTRACT:
0000    33  ;
0000    34  ; The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000    35  ; modules for input to the VAX-11 LINKER.
0000    36  ;
0000    37  ; ENVIRONMENT:  USER MODE
0000    38  ;
0000    39  ; AUTHOR: Benn Schreiber, CREATION DATE: 25-AUG-78
0000    40  ;
0000    41  ; MODIFIED BY:
0000    42  ;
0000    43  ;       V03-005         MTR0031         Mike Rhodes             12-Apr-1983
0000    44  ;               Add code to MAC_DEAL_MEM to release dynamic memory structures
0000    45  ;               used for the .LINK directive.
0000    46  ;
0000    47  ;       V03-004         MTR0026         Mike Rhodes             23-Feb-1983
0000    48  ;               Correct the resetting of the related file name size when
0000    49  ;               performing multiple file assemblies.
0000    50  ;
0000    51  ;       V03-003         MTR0021         Mike Rhodes             20-Aug-1982
0000    52  ;               Correct returning of most severe status for single module
0000    53  ;               assemblies containing an error.
0000    54  ;
0000    55  ;       V03-002         MTR0018         Mike Rhodes             7-Jun-1982
0000    56  ;               Add logic to MACRO_EXIT (et. al.) to retain the most severe
0000    57  ;               status of multiple assemblies for the job's exit status.
```

```
0000     58 ;
0000     59 ;       V03-001         MTR0015         Mike Rhodes             7-Jun-1982
0000     60 ;                Add a CLRB W^MAC$INPUT_RLFNM+NAM$B_RSL to reset the related
0000     61 ;                file size, as well as the address.  This corrects a multiple
0000     62 ;                file assembly problem where the second file name is smaller
0000     63 ;                than the first and a related file (such as a library) is
0000     64 ;                present.
0000     65 ;
0000     66 ;       V02-048         BLS0143         Benn Schreiber          6-Feb-1982
0000     67 ;                Change MOVZBL to MOVZWL due to nam$c_maxrss increase.
0000     68 ;
0000     69 ;       V02-047         BLS0057         Benn Schreiber          13-Jun-1981
0000     70 ;                Corrrect addressing modes to general
0000     71 ;
0000     72 ;--
0000     73
```

```
0000     75          .SBTTL  DECLARATIONS
0000     76 ;
0000     77 ; INCLUDE FILES:
0000     78 ;
0000     79
0000     80 ;
0000     81 ; MACROS:
0000     82 ;
0000     83
0000     84          $RABDEF                         ;DEFINE RAB OFFSETS
0000     85          $STSDEF                         ;DEFINE STATUS CONDITION VALUES
0000     86          $CLIDEF                         ;DEFINE CLI OFFSETS (CLI$A_UTILSERV)
0000     87          $MAC_GENVALDEF                  ;DEFINE COMMONLY USED SYMBOLS IN MACRO
0000     88          $MAC_SYMBLKDEF                  ;DEFINE SYMBOL BLOCK OFFSETS
0000     89          $MAC_INTCODDEF                  ;DEFINE INT. FILE OPCODES
0000     90          $MAC_CTLFLGDEF                  ;DEFINE FLAGS IN MAC$GL_FLAGS
0000     91          $MAC_MLFDEF                     ; Define MLF offsets
```

```
                     0177    93          .SBTTL  VAX-11 MACRO ASSEMBLER ENTRY POINT
                     0177    94 ;++
                     0177    95 ; FUNCTIONAL DESCRIPTION:
                     0177    96 ;
                     0177    97 ;       THE ASSEMBLER IS ENTERED AT 'MAC$MACRO_ENTRY'.  THIS ROUTINE
                     0177    98 ;       SETS UP THE CLI CALLBACK ADDRESS, AND THEN PERFORMS
                     0177    99 ;       THE FOLLOWING ACTIONS:
                     0177   100 ;
                     0177   101 ;       1) RESET GLOBAL STORAGE
                     0177   102 ;       2) SET UP STORAGE FOR PASS1
                     0177   103 ;       3) GET A COMMAND AND VALIDATE
                     0177   104 ;       4) PERFORM PASS 1
                     0177   105 ;       5) INITIALIZE STORAGE FOR PASS 2
                     0177   106 ;       6) PERFORM PASS 2
                     0177   107 ;       7) CLOSE FILES AND RETURN TO IMAGE ACTIVATOR
                     0177   108 ;
                     0177   109 ; CALLING SEQUENCE:
                     0177   110 ;
                     0177   111 ;       CALLS   #1,MAC$MACRO_ENTRY
                     0177   112 ;
                     0177   113 ; INPUT PARAMETERS:
                     0177   114 ;
                     0177   115 ;       CLI$A_UTILSERV(AP)      CLI CALL BACK ADDRESS
                     0177   116 ;
                     0177   117 ; IMPLICIT INPUTS:
                     0177   118 ;
                     0177   119 ;       NONE
                     0177   120 ;
                     0177   121 ; OUTPUT PARAMETERS:
                     0177   122 ;
                     0177   123 ;       NONE
                     0177   124 ;
                     0177   125 ; IMPLICIT OUTPUTS:
                     0177   126 ;
                     0177   127 ;       NONE
                     0177   128 ;
                     0177   129 ; COMPLETION CODES:
                     0177   130 ;
                     0177   131 ;       NONE
                     0177   132 ;
                     0177   133 ; SIDE EFFECTS:
                     0177   134 ;
                     0177   135 ;       NONE
                     0177   136 ;
                     0177   137 ;--
                     0177   138
                     0177   139
                     0177   140
                00000000    141          .PSECT  MAC$RO_CODE_COM,NOWRT,GBL,LONG
                     0000   142
                0000 0000   143          .ENTRY  MAC$MACRO_ENTRY,^M<>    ;MACRO-32 ENTRY POINT
                     0002   144
0000'CF    5E   D0  0002   145          MOVL    SP,W^MAC$GL_INI_SP      ;SAVE INITIAL SP FOR ERROR RECOVERY
0000'CF    5C   D0  0007   146          MOVL    AP,W^MAC$GL_INI_AP      ;SAVE INITIAL AP FOR ERROR RECOVERY
0000'CF    5D   D0  000C   147          MOVL    FP,W^MAC$GL_INI_FP      ;SAVE INITIAL FP FOR ERROR RECOVERY
5B    0000'CF   9E  0011   148          MOVAB   W^MAC$GL_FLAGS,R11      ;POINT R11 TO THE FLAGS WORD
           6B   7C  0016   149          CLRQ    (R11)                   ;CLEAR ALL FLAGS
```

```
                08 AC   D0  0018   150          MOVL    CLI$A_UTILSERV(AP),-    ;GET CLI CALL BACK ADDRESS
              0000'CF   9E  001B   151                  W^MAC$GL_CLIADDR ; INTO A KNOWN LOCATION
00000000'GF     01   9A  001E   152          MOVZBL  #1,G^MAC$GL_FNLSTS      ;ASSUME A SUCCESSFUL ASSEMBLY
                     0025   153 GET_CMD:
              0000'CF   9F  0025   154          PUSHAB  W^MAC$GQ_RNT_TOT        ;STACK TIMING BLOCK ADDRESS
        0000'CF   01   FB  0029   155          CALLS   #1,W^MAC$TIMER_ON       ;BEGIN TIMING WHOLE ASSEMBLER RUN
              0000'CF   9F  002E   156          PUSHAB  W^MAC$GQ_RNT_INI        ;STACK TIMING BLOCK ADDRESS
        0000'CF   01   FB  0032   157          CALLS   #1,W^MAC$TIMER_ON       ;BEGIN TIMING INITIALIZATION
                     0037   158          $CREATE FAB=W^MAC$TERM_FAB,-    ;CREATE TERMINAL OUTPUT CHANNEL
                     0037   159                  ERR=W^MAC$ERR_OPN_OUT
             37 50   E9  0046   160          BLBC    R0,MAC$LAST_CHANCE      ;BRANCH IF ERROR
                     0049   161          $CONNECT RAB=W^MAC$TERM_RAB,-   ;CONNECT THE RECORD STREAM
                     0049   162                  ERR=W^MAC$ERR_OPN_OUT
             25 50   E9  0058   163          BLBC    R0,MAC$LAST_CHANCE      ;BRANCH IF ERROR
                     005B   164 ;
                     005B   165 ; GET A COMMAND AND PROCESS IT
                     005B   166 ;
                     005B   167 10$:
        0000'CF   5E   D0  005B   168          MOVL    SP,W^MAC$GL_SAVE_SP     ;SAVE STACK POINTER FOR ERROR RECOVERY
              00A9   30  0060   169          BSBW    MAC$SETUP               ;SET UP TO PROCESS A COMMAND
              0000'CF   9F  0063   170          PUSHAB  W^MAC$GQ_RNT_INI        ;STACK TIMING BLOCK ADDRESS
        0000'CF   01   FB  0067   171          CALLS   #1,W^MAC$TIMER_OFF      ;STOP TIMING INITIALIZATION
                FF91'   30  006C   172          BSBW    MAC$GETCMD              ;PARSE A COMMAND LINE
                0282    30  006F   173          BSBW    MAC$INITP1              ;INITIALIZE FOR PASS 1
                02D0    30  0072   174          BSBW    MAC$PASS1               ;PERFORM PASS 1 ON THE INPUT
                022F    30  0075   175          BSBW    MAC$INITP2              ;INITIALIZE FOR PASS 2
                FF85'   30  0078   176          BSBW    MAC$PASS2_DRIVR         ;PERFORM PASS 2
                FF82'   30  007B   177          BSBW    MAC$CLOSE_FILES         ;CLOSE OUTPUT FILES
                  1A    11  007E   178          BRB     MACRO_EXIT              ;GO EXIT
                     0080   179
                     0080   180 MAC$LAST_CHANCE::
           5B   0000'CF   9E  0080   181          MOVAB   W^MAC$GL_FLAGS,R11      ;RESET R11 TO POINT TO FLAGS
                     0085   182                                          ;(MAY HAVE BEEN WIPED)
                  FF78'   30  0085   183          BSBW    MAC$CLS_DEL_OBJ         ;DELETE OBJECT FILE IF IT EXISTS
                  FF75'   30  0088   184          BSBW    MAC$CLOSE_FILES         ;CLOSE THE REST OF THE FILES
           5E   0000'CF   D0  008B   185          MOVL    W^MAC$GL_INI_SP,SP      ;RESET SP
           5C   0000'CF   D0  0090   186          MOVL    W^MAC$GL_INI_AP,AP      ;AND AP
           5D   0000'CF   D0  0095   187          MOVL    W^MAC$GL_INI_FP,FP      ;AND FP
                     009A   188 MACRO_EXIT:
                     009A   189          $DISCONNECT RAB=W^MAC$TERM_RAB  ;DISCONNECT TERMINAL I/O
                     00A5   190          $CLOSE  FAB=W^MAC$TERM_FAB      ;AND CLOSE THE FILE
         FFFFFFF8 8F   CB  00B0   191          BICL3   #^CSTS$M_SEVERITY,-     ;GET THE SEVERITY BITS FROM THE CURRENT
      50   00000000'GF       00B6   192                  G^MAC$GL_STATUS, R0      ; ASSEMBLY STATUS.
              1D 50   E8  00BC   193          BLBS    R0, 10$                 ;WARNINGS OR ERRORS INDICATED?
50 00000000'GF   03   00   ED  00BF   194          CMPZV   #0, #3, G^MAC$GL_FNLSTS, R0 ;WE KNOW THERE IS A WARNING/ERROR.
                  07   1F  00C8   195          BLSSU   5$                      ;IS THERE A CHANGE IN STATUS?
        0B 00000000'GF   E9  00CA   196          BLBC    G^MAC$GL_FNLSTS, 10$    ;NO--TRAP SUCCESS vs. WARNING CONDITION.
00000000'GF   00000000'GF   D0  00D1   197 5$:      MOVL    G^MAC$GL_STATUS, G^MAC$GL_FNLSTS ;UPDATE THE EXIT STATUS.
        12 0000'CF   23   E0  00DC   198 10$:     BBS     #FLG$V_MOREINP,W^MAC$GL_FLAGS,20$ ;BRANCH IF MORE INPUT FILES
           50   0000'CF   D0  00E2   199          MOVL    W^MAC$GL_FNLSTS,R0      ;GET THE FINAL STATUS
             00 50   1C   E3  00E7   200          BBCS    #STS$V_INHIB_MSG,R0,.+1 ;DO NOT REPRINT THE ERROR MESSAGE
                     00EB   201          $EXIT_S R0                      ;EXIT WITH CODE IN R0
                0004'DF   D4  00F4   202 20$:     CLRL    @W^MAC$GL_INTQUE+4      ;ZERO LINK IN LAST INTERMEDIATE BUFFER
                0000'CF   90  00F8   203          MOVB    W^MAC$GB_INPNAMLEN,-    ;RESET THE RELATED FILE SIZE
                0003'CF       00FC   204                  W^MAC$INPUT_RLFNM+NAM$B_RSL
                0000'CF   9E  00FF   205          MOVAB   W^MAC$INP_NAM_BUF,-     ;AND ADDRESS FIELDS.
                0004'CF       0103   206                  W^MAC$INPUT_RLFNM+NAM$L_RSA
```

```
00F1   30  0106   207          BSBW    MAC_DEAL_MEM            ;DEALLOCATE DYNAMIC MEMORY STRUCTURES
FF19   31  0109   208          BRW     GET_CMD                 ;GO GET THE NEXT INPUT FILE
```

```
                                010C   210          .SBTTL  SETUP GLOBAL STORAGE TO PROCESS A COMMAND
                                010C   211
                                010C   212 ;++
                                010C   213 ; FUNCTIONAL DESCRIPTION:
                                010C   214 ;
                                010C   215 ;       THIS ROUTINE INITIALIZES GLOBAL STORAGE IN PREPARATION
                                010C   216 ;       FOR PROCESSING A COMMAND LINE.
                                010C   217 ;
                                010C   218 ;--
                                010C   219
                                010C   220 MAC$SETUP:
0000'8F   00   6E   00   2C     010C   221          MOVC5   #0,(SP),#0,#MAC$GK_IMP_SIZ,W^MAC$GL_IMP_BEG ;CLEAR
                    0000'CF     0113
                                0116   222                                              ;IMPURE STORAGE
0200 8F   00   6E   00   2C     0116   223          MOVC5   #0,(SP),#0,#<HASHSZ+1>*4,- ;ZERO THE USER SYMBOL HASH TABLE
                    0000'CF     011D   224                  W^MAC$AL_USYHSHTB
0200 8F   00   6E   00   2C     0120   225          MOVC5   #0,(SP),#0,#<HASHSZ+1>*4,- ;ZERO THE USER MACRO HASH TABLE
                    0000'CF     0127   226                  W^MAC$AL_UMCHSHTB
                                012A   227 ;
                                012A   228 ; Translate logical name SYS$LP_LINES to get lines/page value.
                                012A   229 ;
      00000000'GF   00   FB     012A   230          CALLS   #0,G^LIB$LP_LINES       ; Get number of lines
0000'CF   50   09   C3          0131   231          SUBL3   #9,R0,W^MAC$GL_LN_PAGE  ; Set size allowing for 3 line top
                                0137   232                                          ; margin, 3 line bottom margin and
                                0137   233                                          ; 3 lines for header
                                0137   234 ;
                                0137   235 ; INITIALIZE LISTING HEADER BUFFER
                                0137   236 ;
0000'8F   20   6E   00   2C     0137   237          MOVC5   #0,(SP),#^A/ /,#MAC$K_HD_SIZE,- ;SET BUFFER TO SPACES
                    0000'CF     013E   238                  W^MAC$AB_HD_TITLE
0000'8F   20   6E   00   2C     0141   239          MOVC5   #0,(SP),#^A/ /,#MAC$K_SBT_SIZ,- ;SET SUBTITLE BUFFER TO SPACES
                    0000'CF     0148   240                  W^MAC$AB_SBT_IDNT
          50   0000'CF   9E     014B   241          MOVAB   W^MAC$AB_VERSION,R0     ; Get address of version string
               51   80   9A     0150   242          MOVZBL  (R0)+,R1                ;GET LENGTH OF VERSION STRING
     0000'CF   60   51   28     0153   243          MOVC3   R1,(R0),W^MAC$AB_HD_VERSN ;COPY VERSION INTO BUFFER
          50   0000'CF   9E     0159   244          MOVAB   W^MAC$AB_DEF_TITL,R0    ; Point to default title
               51   80   9A     015E   245          MOVZBL  (R0)+,R1                ;GET LENGTH OF DEFAULT TITLE
     0000'CF   60   51   28     0161   246          MOVC3   R1,(R0),W^MAC$AB_HD_TITLE ;SET AS DEFAULT HD_TITLE
                                0167   247          $ASCTIM_S TIMBUF=W^MAC$AL_ATIM_DSC ; Set time into buffer
          50   0000'CF   9E     0178   248          MOVAB   W^MAC$AB_HD_PAGE,R0     ;POINT TO WHERE PAGE GOES
80   30202020 65676150   8F     7D     017D   249          MOVQ    #^A/Page   0/,(R0)+     ; Store "Page   0"
        6B   00000040   8F     C8     0188   250          BISL2   #FLG$M_EVALEXPR,(R11)   ;SET EVALUATE EXPRESSION
           0000'CF   01   D0     018F   251          MOVL    #1,W^MAC$GL_LSB         ;START IN LSB 1
         0000'CF   FF   8F     98     0194   252          CVTBL   #-1,W^MAC$GL_LIST_IT    ;ASSUME LISTING
        0000'CF   7530   8F     B0     019A   253          MOVW    #30000.,W^MAC$GL_CRSYM  ;START CREATED SYMBOLS AT 30000.
                    02   90     01A1   254          MOVB    #RDX$V_DECIMAL,-        ;SET RADIX TO DECIMAL
                    0000'CF     01A3   255                  W^MAC$GB_RDXNDX
          50   0000'CF   9E     01A6   256          MOVAB   W^MAC$GL_INTQUE,R0      ;INIT THE INT. FILE QUEUE
               60   50   D0     01AB   257          MOVL    R0,(R0)                 ;...
               60   80   DE     01AE   258          MOVAL   (R0)+,(R0)              ;...
          50   0000'CF   9E     01B1   259          MOVAB   W^MAC$GL_INPQUE,R0      ;INIT THE INPUT FILE QUEUE
               60   50   D0     01B6   260          MOVL    R0,(R0)                 ;...
               60   80   DE     01B9   261          MOVAL   (R0)+,(R0)              ;...
          50   0000'CF   9E     01BC   262          MOVAB   W^MAC$GL_ERR_LIST,R0    ;INIT THE ERROR LIST QUEUE
               60   50   D0     01C1   263          MOVL    R0,(R0)                 ;...
               60   80   DE     01C4   264          MOVAL   (R0)+,(R0)              ;...
          50   0000'CF   9E     01C7   265          MOVAB   W^MAC$GL_FREE_LST,R0    ;INIT THE FREE PAGES LIST
```

```
                60    50    D0   01CC    266          MOVL    R0,(R0)                 ;...
                60    80    DE   01CF    267          MOVAL   (R0)+,(R0)              ;...
          50    0000'CF     9E   01D2    268          MOVAB   W^PSECT$BLANK,R0        ;POINT TO THE BLANK PSECT
0000'CF         50    D0   01D7    269          MOVL    R0,W^MAC$GL_PSECTPTR    ;START POINTER IN DEFAULT PSECT
              0F  A0    D4   01DC    270          CLRL    PSC$L_CURLOC(R0)        ;START AT 0
              05  A0    D4   01DF    271          CLRL    PSC$L_MAXLGTH(R0)       ;...
          50    0000'CF     9E   01E2    272          MOVAB   W^MAC$GL_SYM_PAGL,R0    ;INIT THE SYMBOL PAGES QUEUE
                60    50    D0   01E7    273          MOVL    R0,(R0)                 ;...
                60    80    DE   01EA    274          MOVAL   (R0)+,(R0)              ;...
0000'CF         01    9A   01ED    275          MOVZBL  #1,W^MAC$GL_PSECT       ;PSECT 1
0000'CF         01    9A   01F2    276          MOVZBL  #1,W^MAC$GL_PSC_MAX     ;START WITH 1
              FE06'   31   01F7    277          BRW     MAC$SYSLIB_SET          ;SET UP SYSTEM MACRO LIBRARY AND RETURN
```

```
                        01FA   279          .SBTTL  DEALLOCATE DYNAMIC MEMORY STRUCTURES
                        01FA   280
                        01FA   281 ;++
                        01FA   282 ; FUNCTIONAL DESCRIPTION:
                        01FA   283 ;
                        01FA   284 ;       THIS ROUTINE IS CALLED IF THERE ARE MULTIPLE ASSEMBLIES TO
                        01FA   285 ;       DEALLOCATE ALL DYNAMIC MEMORY STRUCTURES.
                        01FA   286 ;
                        01FA   287 ;       (REGISTERS NOT SAVED--IN BETWEEN ASSEMBLIES)
                        01FA   288 ;--
                        01FA   289
                        01FA   290 MAC_DEAL_MEM:
                        01FA   291 ;
                        01FA   292 ; DEALLOCATE SYMBOL PAGES
                        01FA   293 ;
 50  0000'DF   0F   01FA   294 10$:    REMQUE  @W^MAC$GL_SYM_PAGL,R0   ;GET NEXT CHUNK OF PAGES TO DEALLOCATE
        0A   1D   01FF   295           BVS     20$                     ;IF V-SET THEN ALL DONE
 51  1400 8F   3C   0201   296           MOVZWL  #<512*STB$K_PG_MISS>,R1 ;GET SIZE OF CHUNK
     0085   30   0206   297           BSBW    DEAL_MEMORY             ;DEALLOCATE THE MEMORY
        EF   11   0209   298           BRB     10$                     ;FREE ALL SYMBOL PAGES
                        020B   299 ;
                        020B   300 ; NOW DEALLOCATE THE INTERMEDIATE FILE
                        020B   301 ;
 52  0000'CF   D0   020B   302 20$:    MOVL    W^MAC$GL_INTQUE,R2      ;POINT AT THE INTERMEDIATE FILE
     50   52   D0   0210   303 30$:    MOVL    R2,R0                   ;ANY MORE BLOCKS?
        0D   13   0213   304           BEQL    40$                     ;IF EQL NO
     52   62   D0   0215   305           MOVL    (R2),R2                 ;YES--LINK TO NEXT
 51  13F4 8F   3C   0218   306           MOVZWL  #INT$K_BUFSIZ,R1        ;GET SIZE OF THE BLOCK
     006E   30   021D   307           BSBW    DEAL_MEMORY             ;DEALLOCTE THE BLOCK
        EE   11   0220   308           BRB     30$                     ;DEALLOCATE WHOLE INTER. FILE
                        0222   309 ;
                        0222   310 ; DEALLOCATE ANY MACROS DEFINED
                        0222   311 ;
 59  0000'CF   9E   0222   312 40$:    MOVAB   W^MAC$AL_UMCHSHTB,R9    ;POINT TO MACRO HASH TABLE
 58  0080 8F   3C   0227   313           MOVZWL  #<HASHSZ+1>,R8          ;COUNT OF THE ENTRIES
     57   89   D0   022C   314 50$:    MOVL    (R9)+,R7                ;GET NEXT BUCKET POINTER
        0D   13   022F   315           BEQL    70$                     ;IF EQL NONE
     56   57   D0   0231   316 60$:    MOVL    R7,R6                   ;SET POINTER INTO R6
        08   13   0234   317           BEQL    70$                     ;IF EQL NO MORE
     57   66   D0   0236   318           MOVL    (R6),R7                 ;GET POINTER TO NEXT MNB OR 0
        FDC4'   30   0239   319           BSBW    MAC$DEL_MAC_DEF         ;DELETE THE MACRO DEF.
        F3   11   023C   320           BRB     60$                     ;CONTINUE DELETING ON THIS BUCKET
     EB 58   F5   023E   321 70$:    SOBGTR  R8,50$                  ;DELETE ALL MACRO DEFS.
                        0241   322 ;
                        0241   323 ; DELETE THE FREE PAGES LIST
                        0241   324 ;
 50  0000'DF   0F   0241   325 80$:    REMQUE  @W^MAC$GL_FREE_LST,R0   ;GET A PAGE
        0A   1D   0246   326           BVS     90$                     ;IF V-SET NO MORE
 51  0000'CF   3C   0248   327           MOVZWL  W^MAC$GK_1_PG_SIZ,R1    ;GET SIZE OF PAGE
     003E   30   024D   328           BSBW    DEAL_MEMORY             ;DEALLOCATE THE PAGE
        EF   11   0250   329           BRB     80$                     ;CONTINUE
                        0252   330 ;
                        0252   331 ; DEALLOCATE THE MACRO LIBRARY QUEUE AND THE INPUT FILE QUEUE
                        0252   332 ;
 50  0000'DF   0F   0252   333 90$:    REMQUE  @W^MAC$GL_MLB_QUE,R0    ;GET NEXT MLB TO RELEASE
        13   1D   0257   334           BVS     100$                    ;IF VS NO MORE
00000000'8F   50   D1   0259   335           CMPL    R0,#MAC$SYSLIB_MLF      ;IS THIS SYSLIB?
```

```
                     0A   13   0260   336          BEQL    100$                        ;IF EQL YES--WE ARE DONE
           51  0177 8F   3C   0262   337          MOVZWL  #MLF$K_BLKSIZ,R1            ;FIGURE BLOCK SIZE
                 0024   30   0267   338          BSBW    DEAL_MEMORY                 ;DEALLOCATE IT
                   E6   11   026A   339          BRB     90$
           50  0000'DF   0F   026C   340 100$:    REMQUE  @W^MAC$GL_INPQUE,R0         ;GET NEXT INPUT FILE BLOCK
                   0A   1D   0271   341          BVS     110$                        ;IF V-SET NO MORE
           51  0000'CF   3C   0273   342          MOVZWL  W^MAC$GK_1_PG_SIZ,R1        ;BLOCK IS ONE PAGE
                 0013   30   0278   343          BSBW    DEAL_MEMORY                 ;DEALLOCATE IT
                   EF   11   027B   344          BRB     100$                        ;DO THEM ALL
                        027D   345 ;
                        027D   346 ; DEALLOCATE THE LINKER OPTION RECORD(S).
                        027D   347 ;
           50  0000'DF   0F   027D   348 110$:    REMQUE  @W^MAC$GQ_LNKOPT,R0         ;GET THE LINKER OPTION RECORD'S ADDRESS.
                   09   1D   0282   349          BVS     120$                        ;IS THE QUEUE EMPTY?
             51  08 A0   D0   0284   350          MOVL    8(R0),R1                    ;NO, GET THE BLOCK SIZE
                 0003   30   0288   351          BSBW    DEAL_MEMORY                 ;AND DEALLOCATE IT.
                   F0   11   028B   352          BRB     110$                        ;EMPTY THE QUEUE.
                        028D   353
                   05   028D   354 120$:    RSB
                        028E   355
                        028E   356 ;++
                        028E   357 ; ROUTINE TO DEALLOCATE MEMORY.
                        028E   358 ;
                        028E   359 ;       R0 - ADDRESS OF BLOCK
                        028E   360 ;       R1 - SIZE
                        028E   361 ;--
                        028E   362
                        028E   363 DEAL_MEMORY:
                50   DD   028E   364          PUSHL   R0                          ;STACK BLOCK ADDRESS
              50   5E   D0   0290   365          MOVL    SP,R0                       ;REMEMBER ITS ADDRESS
                51   DD   0293   366          PUSHL   R1                          ;STACK BLOCK SIZE
              51   5E   D0   0295   367          MOVL    SP,R1                       ;REMEMBER ITS ADDRESS
                50   DD   0298   368          PUSHL   R0                          ;STACK ADDRESS OF ADDRESS
                51   DD   029A   369          PUSHL   R1                          ;AND THE SIZE
00000000'GF     02   FB   029C   370          CALLS   #2,G^LIB$FREE_VM            ;FREE THE MEMEORY
              5E   08   C0   02A3   371          ADDL2   #2*4,SP                     ;CLEAR THE STACK
                   05   02A6   372          RSB
```

```
                     02A7   374          .SBTTL  INITIALIZE FOR ONE PASS THROUGH THE SOURCE
                     02A7   375
                     02A7   376 ;++
                     02A7   377 ; FUNCTIONAL DESCRIPTION:
                     02A7   378 ;
                     02A7   379 ;       THESE ROUTINES INITIALIZE THE STORAGE FOR ONE PASS THROUGH
                     02A7   380 ;       THE SOURCE.
                     02A7   381 ;
                     02A7   382 ; CALLING SEQUENCE:
                     02A7   383 ;
                     02A7   384 ;       JSB     MAC$INITP1
                     02A7   385 ; OR    JSB     MAC$INITP2
                     02A7   386 ;
                     02A7   387 ; INPUT PARAMETERS:
                     02A7   388 ;
                     02A7   389 ;       NONE
                     02A7   390 ;
                     02A7   391 ; IMPLICIT INPUTS:
                     02A7   392 ;
                     02A7   393 ;       FOR MAC$INITP1 IT IS EXPECTED THAT MAC$GL_FLAGS HAS BEEN
                     02A7   394 ;       ZEROED AND THAT R11 POINTS TO THE FLAGS.
                     02A7   395 ;
                     02A7   396 ;--
                     02A7   397
                     02A7   398 MAC$INITP2:                             ;ENTRY POINT FOR PASS 2 INITIALIZATION
     50  0000'CF    D0  02A7   399          MOVL    W^MAC$GL_SYMPGPTR,R0    ;GET POINTER TO LAST PAGES ALLOCATED
           05        13  02AC   400          BEQL    2$                      ;IF EQL NONE ALLOCATED
0000'CF  60        0E  02AE   401          INSQUE  (R0),W^MAC$GL_SYM_PAGL  ;LINK LAST PAGES INTO SYMBOL PAGE QUEUE
     59  0000'CF    9E  02B3   402 2$:     MOVAB   W^MAC$AL_USYHSHTB,R9    ;POINT TO USER SYMBOL HASH TABLE
     58  0080 8F    3C  02B8   403          MOVZWL  #HASHSZ+T,R8            ;LOAD UP SIZE OF TABLE
             FD40'   30  02BD   404          BSBW    MAC$SORT_TABLE          ;SORT THE SYMBOL TABLE
           0000'CF   9F  02C0   405          PUSHAB  W^MAC$GQ_RNT_P2         ;STACK TIMING BLOCK ADDRESS
      0000'CF  01    FB  02C4   406          CALLS   #1,W^MAC$TIMER_ON       ;START TIMING PASS 2
        00 6B  0E    E3  02C9   407          BBCS    #FLG$V_P2,(R11),10$      ;FLAG PASS 2 IS UP
0000'8F  20  6E  00    2C  02CD   408 10$:    MOVC5   #0,(SP),#^A/ /,#MAC$K_SBT_SIZ,- ;BLANK FILL SUBTITLE BUFFER
           0000'CF             02D4   409                  W^MAC$AB_SBT_IDNT
           56  0000'CF    9E  02D7   410          MOVAB   W^MAC$AB_IDENT,R6       ;POINT TO IDENT STORAGE
               57  86      9A  02DC   411          MOVZBL  (R6)+,R7                ;GET LENGTH OF IDENT
                   1F      13  02DF   412          BEQL    INIT_0                  ;IF EQL NO IDENT
     0000'CF  66  57    28  02E1   413          MOVC3   R7,(R6),W^MAC$AB_SBT_IDNT ;COPY IDENT INTO SUBTITLE BUFFER
     0000'CF  57  09    3A  02E7   414 20$:    LOCC    #TAB,R7,W^MAC$AB_SBT_IDNT ;FIND ANY TABS?
                   11      13  02ED   415          BEQL    INIT_0                  ;IF EQL NO
               61  20      90  02EF   416          MOVB    #^A/ /,(R1)             ;YES--CHANGE TO SPACE
                   F3      11  02F2   417          BRB     20$                     ;CHANGE ALL THE TABS
                           02F4   418
                           02F4   419 MAC$INITP1:                             ;REF LABEL
           0000'CF   9F  02F4   420          PUSHAB  W^MAC$GQ_RNT_P1         ;STACK TIMING BLOCK ADDRESS
      0000'CF  01    FB  02F8   421          CALLS   #1,W^MAC$TIMER_ON       ;START TIMING PASS 1
             FD00'   30  02FD   422          BSBW    MAC$SETFRAME            ;GET BLOCK OF MEMORY AND SETUP
                           0300   423                                          ;TO STORE IN INT. BUFFER (SETUP R9)
                           0300   424
                           0300   425 INIT_0:
           0000'CF   D4  0300   426          CLRL    W^MAC$GL_LIST_LVL       ;START LISTING AT LEVEL 0
           0000'CF   D4  0304   427          CLRL    W^MAC$GL_LINE_CNT       ;ZERO PAGE LINE COUNTER
           0000'CF   D4  0308   428          CLRL    W^MAC$GL_LPTPAG         ;FIRST LISTING PAGE NUMBER
      0000'CF  01    9A  030C   429          MOVZBL  #1,W^MAC$GL_SRCPAG      ;FIRST SOURCE PAGE NUMBER
           0000'CF   D4  0311   430          CLRL    W^MAC$GL_LINENUM        ;FIRST LINE
```

```
                0000'CF   D4   0315   431          CLRL    W^MAC$GL_LINBAS           ;INIT LINE BASE ALSO
                6B    08  C8   0319   432          BISL2   #FLG$M_CONT,(R11)         ;INDICATE CONTINUATION OK
             00 6B    0A  E5   031C   433          BBCC    #FLG$V_NEWPND,(R11),10$ ;NEW PAGE NOT NEEDED
          50    0000'CF   9E   0320   434 10$:     MOVAB   W^MAC$GL_PRMINBL,R0       ;POINT TO PRIMARY INPUT BLOCK
                51    50  D0   0325   435          MOVL    R0,R1                     ;COPY IT
          0000'CF   51    D0   0328   436          MOVL    R1,W^MAC$GL_INPUTP        ;SET UP INPUT POINTER
                80    50  D0   032D   437          MOVL    R0,(R0)+                  ;LINK IS TO ITSELF
                      80  D4   0330   438          CLRL    (R0)+                     ;THERE IS NO NEXT LINE
          80    0000'CF   9E   0332   439          MOVAB   W^MAC$GETLIN,(R0)+        ;SET ROUTINE TO GET NEXT LINE
                      80  7C   0337   440          CLRQ    (R0)+                     ;CLEAR IFLVL AND IFVAL
                      80  D4   0339   441          CLRL    (R0)+                     ;CLEAR PAGE POINTER
                      80  94   033B   442          CLRB    (R0)+                     ;CLEAR ARG COUNT
          50    0000'CF   D0   033D   443          MOVL    W^MAC$GL_INPQUE,R0        ;GET PTR TO FIRST FDB IN INP. QUEUE
                   FCBB'  31   0342   444          BRW     MAC$OPEN_INPUT            ;OPNE FILE AND RETURN
```

```
                     0345   446          .SBTTL  PERFORM PASS 1
                     0345   447
                     0345   448 ;++
                     0345   449 ;
                     0345   450 ;       PASS 1
                     0345   451 ;
                     0345   452 ;--
                     0345   453
                     0345   454 MAC$PASS1:
   00 6B   0E   E5   0345   455          BBCC    #FLG$V_P2,(R11),.+1     ;THIS IS PASS 1
                     0349   456 ;
                     0349   457 ; COPY THE INITIAL SETTINGS OF THE ENABLE/DISABLE AND LIST/NLIST
                     0349   458 ; FLAGS TO THE TOKEN BYTE IN EACH OF THE SYMBOL BLOCKS SO THEY
                     0349   459 ; CAN BE RESET AT THE START OF PASS 2.
                     0349   460 ;
 55  0000'CF   9E   0349   461          MOVAB   W^LST$G_DIRLIST,R5      ;POINT TO DIRECTIVE LIST
0B A5   05 A5   90   034E   462 10$:     MOVB    SYM$L_VAL(R5),SYM$B_TOKEN(R5) ;SAVE THE INITIAL SETTING
      55   65   D0   0353   463          MOVL    SYM$L_LINK(R5),R5       ; Link to next
           F6   12   0356   464          BNEQ    10$
 55  0000'CF   9E   0358   465          MOVAB   W^ENB$G_OPTIONS,R5      ;POINT TO ENABLE OPTIONS
0B A5   05 A5   90   035D   466 20$:     MOVB    SYM$L_VAL(R5),SYM$B_TOKEN(R5) ;SAVE INITIAL SETTING
      55   65   D0   0362   467          MOVL    SYM$L_LINK(R5),R5       ; Link to next
           F6   12   0365   468          BNEQ    20$                     ;LOOP FOR ALL
                     0367   469          $INTOUT_LW INT$_NEWP,#PSECT$MAIN ;ABSOLUTE PSECT
                     0373   470          $INTOUT_LW INT$_NEWP,#PSECT$BLANK ;BLANK PSECT
                     037F   471          $INTOUT_LW INT$_PSECT,#PSECT$BLANK ;START IN BLANK PSECT
0000'CF   5E   D0   038B   472          MOVL    SP,W^MAC$GL_SAVE_SP     ;SAVE STACK POINTER
         FC6D'  30   0390   473          BSBW    MAC$PARSE               ;CALL PASS 1 DRIVER
                     0393   474 ;
                     0393   475 ; PASS 1 IS COMPLETED
                     0393   476 ;
                     0393   477 ;
                     0393   478 ; THE ROUTINE MAC$PARSE DOES NOT RETURN.  RATHER, WHEN THE END
                     0393   479 ; STATEMENT IS SEEN (OR FORCED), CONTROL WILL COME TO MAC$PASS1_END
                     0393   480 ; FOR A NORMAL END OF PASS 1 OR TO MAC$ABORT_PASS1 IF IT IS
                     0393   481 ; ABORTED.
                     0393   482 ;
                     0393   483 MAC$ABORT_PASS1::
         FC6A'  30   0393   484          BSBW    MAC$CLS_DEL_OBJ         ;CLOSE AND DELETE OBJECT FILE
                     0396   485 MAC$PASS1_END::
   5E  0000'CF   D0   0396   486          MOVL    W^MAC$GL_SAVE_SP,SP     ;RESTORE STACK POINTER
                     039B   487          $INTOUT_X INT$_END              ;END OF INTERMEDIATE FILE
         FC5C'  30   03A1   488          BSBW    MAC$FIXFRAME            ;FIX THE COUNT WORD IN LAST BUFFER
                     03A4   489                                          ; TO VIRTUAL MEMORY
         0004'DF   D4   03A4   490          CLRL    @W^MAC$GL_INTQUE+4      ;ZERO LINK IN LAST BUFFER
                     03A8   491                                          ; SO THAT PASS 2 CAN DETECT ERROR
         0000'CF   9F   03A8   492          PUSHAB  W^MAC$GT_SCB            ; Suplly SUM control block address
00000000'GF   01   FB   03AC   493          CALLS   #1,G^SUM$CLOSE          ; Close update files
                     03B3   494          $DISCONNECT RAB=W^MAC$INPUT_RAB ;DISCONNECT THE RECORD STREAM
   50  0000'CF   D0   03BE   495          MOVL    W^MAC$GL_CURINFDB,R0    ;POINT TO CURRENT INPUT FDB
                     03C3   496          $CLOSE  FAB=8(R0)               ;CLOSE THE INPUT FILE
         FC30'  30   03CD   497          BSBW    MAC$CLOSE_LIB           ;CLOSE MACRO LIBRARY FILES
         0000'CF   9F   03D0   498          PUSHAB  W^MAC$GQ_RNT_P1         ;STACK TIMING BLOCK ADDRESS
0000'CF   01   FB   03D4   499          CALLS   #1,W^MAC$TIMER_OFF      ;STOP TIMING PASS 1
                05   03D9   500          RSB                             ;PASS 1 IS COMPLETED
                     03DA   501
                     03DA   502          .END    MAC$MACRO_ENTRY
```

```
$$.TMP1                         = 00000001
$$.TMP2                         = 000000A0
$COUNT                          = 0000003B
ARG$K_SIZE                      = 000003E8
AUD$K_SIZE                      = 00000010
BLNK                            = 00000020
CHR$M_COMMA_CR                  = 00000020
CHR$M_ILL_CHR                   = 00000040
CHR$M_NUM_BER                   = 00000010
CHR$M_SPA_MSK                   = 00000001
CHR$M_SYM_CH1                   = 00000008
CHR$M_SYM_CHR                   = 00000004
CHR$M_SYM_DLM                   = 00000002
CHR$V_COMMA_CR                  = 00000005
CHR$V_CVTLWC                    = 00000061
CHR$V_ILL_CHR                   = 00000006
CHR$V_NOCVT                     = 0000007F
CHR$V_NUM_BER                   = 00000004
CHR$V_SPA_MSK                   = 00000000
CHR$V_SYM_CH1                   = 00000003
CHR$V_SYM_CHR                   = 00000002
CHR$V_SYM_DLM                   = 00000001
CLI$A_UTILSERV                  = 00000008
CNT                             = 00000001
CR                              = 0000000D
DEAL_MEMORY                       0000028E R     03
ENB$G_OPTIONS                     ******** X     03
ERR                             = 00000000
FF                              = 0000000C
FLG$M_ALLCHR                    = 00000001
FLG$M_BOL                       = 00000002
FLG$M_CHKLPND                   = 00100000
FLG$M_COMPEXPR                  = 00000004
FLG$M_CONT                      = 00000008
FLG$M_CRF                       = 40000000
FLG$M_CRSEEN                    = 00000001
FLG$M_DATRPT                    = 00000010
FLG$M_DBGOUT                    = 00004000
FLG$M_DLIMSTR                   = 00008000
FLG$M_ENDMCH                    = 00000020
FLG$M_EVALEXPR                  = 00000040
FLG$M_EXPOPT                    = 00000080
FLG$M_EXTERR                    = 00010000
FLG$M_EXTWRN                    = 00020000
FLG$M_FIRSTLN                   = 00000200
FLG$M_IFSTAT                    = 00800000
FLG$M_IIF                       = 00400000
FLG$M_INSERT                    = 00000100
FLG$M_IRPC                      = 20000000
FLG$M_LEXOP                     = 00000002
FLG$M_LSTXST                    = 00000200
FLG$M_MAC2COL                   = 00000800
FLG$M_MACL                      = 00000800
FLG$M_MACLTB                    = 08000000
FLG$M_MACTXT                    = 00010000
FLG$M_MEBLST                    = 00001000
FLG$M_MOREARG                   = 00002000
FLG$M_MOREINP                   = 00000008
FLG$M_NEWPND                    = 00000400
FLG$M_NOREF                     = 01000000
FLG$M_NTYPEPC                   = 00000020
FLG$M_NULCHR                    = 00040000
FLG$M_OBJXST                    = 00200000
FLG$M_OPNDCHK                   = 00000100
FLG$M_OPRND                     = 00002000
FLG$M_OPTVFLIDX                 = 00001000
FLG$M_ORDLST                    = 00020000
FLG$M_P2                        = 00004000
FLG$M_RPTIRP                    = 10000000
FLG$M_SEQFIL                    = 02000000
FLG$M_SKAN                      = 00008000
FLG$M_SPECOP                    = 00000004
FLG$M_SPLALL                    = 04000000
FLG$M_STOIMF                    = 00040000
FLG$M_SYM2COL                   = 00000400
FLG$M_TOCFLG                    = 00080000
FLG$M_UPAFLG                    = 00000010
FLG$M_UPDFIL                    = 00000080
FLG$M_UPMARG                    = 0000040C
FLG$M_XCRF                      = 80000000
FLG$V_ALLCHR                    = 00000000
FLG$V_BOL                       = 00000001
FLG$V_CHKLPND                   = 00000014
FLG$V_COMPEXPR                  = 00000002
FLG$V_CONT                      = 00000003
FLG$V_CRF                       = 0000001E
FLG$V_CRSEEN                    = 00000020
FLG$V_DATRPT                    = 00000004
FLG$V_DBGOUT                    = 0000002E
FLG$V_DLIMSTR                   = 0000002F
FLG$V_ENDMCH                    = 00000005
FLG$V_EVALEXPR                  = 00000006
FLG$V_EXPOPT                    = 00000007
FLG$V_EXTERR                    = 00000030
FLG$V_EXTWRN                    = 00000031
FLG$V_FIRSTLN                   = 00000029
FLG$V_IFSTAT                    = 00000017
FLG$V_IIF                       = 00000016
FLG$V_INSERT                    = 00000008
FLG$V_IRPC                      = 0000001D
FLG$V_LEXOP                     = 00000021
FLG$V_LSTXST                    = 00000009
FLG$V_MAC2COL                   = 0000002B
FLG$V_MACL                      = 0000000B
FLG$V_MACLTB                    = 0000001B
FLG$V_MACTXT                    = 00000010
FLG$V_MEBLST                    = 0000000C
FLG$V_MOREARG                   = 0000002D
FLG$V_MOREINP                   = 00000023
FLG$V_NEWPND                    = 0000000A
FLG$V_NOREF                     = 00000018
FLG$V_NTYPEPC                   = 00000025
FLG$V_NULCHR                    = 00000032
FLG$V_OBJXST                    = 00000015
```

```
FLG$V_OPNDCHK                   = 00000028
FLG$V_OPRND                     = 0000000D
FLG$V_OPTVFLIDX                 = 0000002C
FLG$V_ORDLST                    = 00000011
FLG$V_P2                        = 0000000E
FLG$V_RPTIRP                    = 0000001C
FLG$V_SEQFIL                    = 00000019
FLG$V_SKAN                      = 0000000F
FLG$V_SPECOP                    = 00000022
FLG$V_SPLALL                    = 0000001A
FLG$V_STOIMF                    = 00000012
FLG$V_SYM2COL                   = 0000002A
FLG$V_TOCFLG                    = 00000013
FLG$V_UPAFLG                    = 00000024
FLG$V_UPDFIL                    = 00000027
FLG$V_UPMARG                    = 00000026
FLG$V_XCRF                      = 0000001F
GET_CMD                           00000025 R      03
HASASZ                          = 0000007F
HYPHEN                          = 0000002D
INIT_0                            00000300 R      03
INP$K_BUFSIZ                    = 000003E8
INT$K_BUFSIZ                    = 000013F4
INT$K_BUFWRN                    = 00001390
INT$_ADD                        = 00000001
INT$_AND                        = 00000002
INT$_ASH                        = 00000003
INT$_ASN                        = 0000000C
INT$_AUGPC                      = 0000000D
INT$_BDST                       = 0000000E
INT$_CHKL                       = 0000000F
INT$_DIV                        = 00000004
INT$_END                        = 00000010
INT$_EPT                        = 00000011
INT$_ERR                        = 00000012
INT$_ETX                        = 00000013
INT$_FNEWL                      = 00000014
INT$_ILG                        = 00000000
INT$_INFO                       = 0000003A
INT$_LGLAB                      = 00000015
INT$_MACL                       = 00000016
INT$_MUL                        = 00000005
INT$_NEG                        = 00000006
INT$_NEWL                       = 00000017
INT$_NEWP                       = 00000018
INT$_NOT                        = 00000007
INT$_OP                         = 00000019
INT$_OR                         = 00000008
INT$_PRIL                       = 0000001A
INT$_PRT                        = 0000001B
INT$_PSECT                      = 0000001C
INT$_REDEF                      = 0000001D
INT$_REF                        = 0000001E
INT$_REST                       = 0000001F
INT$_SAME                       = 00000009
INT$_SAVE                       = 00000020
INT$_SBTTL                      = 00000021
INT$_SETFLAG                    = 00000022
INT$_SETLONG                    = 00000023
INT$_SPIC                       = 00000024
INT$_SPID                       = 00000025
INT$_STIB                       = 00000026
INT$_STIL                       = 00000028
INT$_STIW                       = 00000027
INT$_STKEPT                     = 00000029
INT$_STKG                       = 0000002A
INT$_STKL                       = 0000002B
INT$_STKPC                      = 0000002C
INT$_STKS                       = 0000002D
INT$_STOB                       = 00000034
INT$_STOL                       = 0000002E
INT$_STOW                       = 00000035
INT$_STRB                       = 0000002F
INT$_STRL                       = 00000031
INT$_STRSB                      = 00000032
INT$_STRSW                      = 00000033
INT$_STRW                       = 00000030
INT$_STSB                       = 00000036
INT$_STSW                       = 00000037
INT$_SUB                        = 0000000A
INT$_SUME                       = 00000039
INT$_WRN                        = 00000038
INT$_XOR                        = 0000000B
LIB$FREE_VM                       ********    X   03
LIB$LP_LINES                      ********    X   03
LST$G_DIRLIST                     ********    X   03
LST$K_BUFSIZ                    = 00000086
LST$K_L_P_PAGE                  = 0000003C
LST$K_TITLE_SIZ                 = 00000028
MAC$ABORT_PASS1                   00000393 RG     03
MAC$AB_DEF_TITL                   ********      X 03
MAC$AB_HD_PAGE                    ********      X 03
MAC$AB_HD_TITLE                   ********      X 03
MAC$AB_HD_VERSN                   ********      X 03
MAC$AB_IDENT                      ********      X 03
MAC$AB_SBT_IDNT                   ********      X 03
MAC$AB_VERSION                    ********      X 03
MAC$AL_ATIM_DSC                   ********      X 03
MAC$AL_UMCHSHTB                   ********      X 03
MAC$AL_USYHSHTB                   ********      X 03
MAC$CLOSE_FILES                   ********      X 03
MAC$CLOSE_LIB                     ********      X 03
MAC$CLS_DEL_OBJ                   ********      X 03
MAC$DEL_MAC_DEF                   ********      X 03
MAC$ERR_OPN_OUT                   ********      X 03
MAC$FIXFRAME                      ********      X 03
MAC$GB_INPNAMLEN                  ********      X 03
MAC$GB_RDXNDX                     ********      X 03
MAC$GETCMD                        ********      X 03
MAC$GETLIN                        ********      X 03
MAC$GK_1_PG_SIZ                   ********      X 03
MAC$GK_IMP_SIZ                    ********      X 03
MAC$GL_CLIADDR                    ********      X 03
MAC$GL_CRSYM                      ********      X 03
```

```
MAC$GL_CURINFDB          ********   X   03
MAC$GL_ERR_LIST          ********   X   03
MAC$GL_FLAGS             ********   X   03
MAC$GL_FNLSTS            ********   X   03
MAC$GL_FREE_LST          ********   X   03
MAC$GL_IMP_BEG           ********   X   03
MAC$GL_INI_AP            ********   X   03
MAC$GL_INI_FP            ********   X   03
MAC$GL_INI_SP            ********   X   03
MAC$GL_INPQUE            ********   X   03
MAC$GL_INPUTP            ********   X   03
MAC$GL_INTQUE            ********   X   03
MAC$GL_LINBAS            ********   X   03
MAC$GL_LINENUM           ********   X   03
MAC$GL_LINE_CNT          ********   X   03
MAC$GL_LIST_IT           ********   X   03
MAC$GL_LIST_LVL          ********   X   03
MAC$GL_LN_PAGE           ********   X   03
MAC$GL_LPTPAG            ********   X   03
MAC$GL_LSB               ********   X   03
MAC$GL_MLB_QUE           ********   X   03
MAC$GL_PRMINBL           ********   X   03
MAC$GL_PSC_MAX           ********   X   03
MAC$GL_PSECT             ********   X   03
MAC$GL_PSECTPTR          ********   X   03
MAC$GL_SAVE_SP           ********   X   03
MAC$GL_SRCPAG            ********   X   03
MAC$GL_STATUS            ********   X   03
MAC$GL_SYMPGPTR          ********   X   03
MAC$GL_SYM_PAGL          ********   X   03
MAC$GQ_LNKOPT            ********   X   03
MAC$GQ_RNT_INI           ********   X   03
MAC$GQ_RNT_P1            ********   X   03
MAC$GQ_RNT_P2            ********   X   03
MAC$GQ_RNT_TOT           ********   X   03
MAC$GT_SCB               ********   X   03
MAC$INITP1               000002F4 R     03
MAC$INITP2               000002A7 R     03
MAC$INPUT_RAB            ********     X 03
MAC$INPUT_RLFNM          ********     X 03
MAC$INP_NAM_BUF          ********     X 03
MAC$INTOUT_T_LW          ********     X 03
MAC$INTOUT_X             ********     X 03
MAC$K_HD_SIZE            ********     X 03
MAC$K_SBT_SIZ            ********     X 03
MAC$LAST_CHANCE          00000080 RG    03
MAC$MACRO_ENTRY          00000000 RG    03
MAC$OPEN_INPUT           ********     X 03
MAC$PARSE                ********     X 03
MAC$PASS1                00000345 R     03
MAC$PASS1_END            00000396 RG    03
MAC$PASS2_DRIVR          ********     X 03
MAC$SETFRAME             ********     X 03
MAC$SETUP                0000010C R     03
MAC$SORT_TABLE           ********     X 03
MAC$SYSLIB_MLF           ********     X 03
MAC$SYSLIB_SET           ********     X 03
MAC$TERM_FAB             ********     X 03
MAC$TERM_RAB             ********     X 03
MAC$TIMER_OFF            ********     X 03
MAC$TIMER_ON             ********     X 03
MACRO_EXIT               0000009A R     03
MAC_DEAL_MEM             0000017A R     03
MAC_SUBSYS             = 0000007D
MLF$K_BLKSIZ             00000177
MLF$K_RSFNLN           = 000000FF
MLF$L_CTINDEX            00000014
MLF$L_MCDEF              00000008
MLF$L_QLINK              00000000
MLF$Q_FNAMDS             0000000C
MLF$T_FNAM               00000078
MLF$X_NAMBLK             00000018
NAM$B_RSL              = 00000003
NAM$C_BLN              = 00000060
NAM$C_MAXRSS           = 000000FF
NAM$L_RSA              = 00000004
OBJ$K_BUFSIZ           = 00000200
OPF$M_LASTOPR          = 00002000
OPF$M_OPTEXP           = 00001000
OPF$V_LASTOPR          = 0000000D
OPF$V_OPTEXP           = 0000000C
PSC$B_NAME               00000004
PSC$B_SEG                0000000C
PSC$B_UNUSED             0000000B
PSC$K_BLKSIZ             00000013
PSC$K_NO_OPTNS         = 0000000A
PSC$L_CURLOC             0000000F
PSC$L_LINK               00000000
PSC$L_MAXLGTH            00000005
PSC$M_ABS              = FFFFFFF7
PSC$M_ALIGNFLG         = 00004000
PSC$M_ALLOPTNS         = 000003FF
PSC$M_BYTE             = 00004000
PSC$M_CON              = FFFFFFFB
PSC$M_DEFAULT          = 000001C8
PSC$M_EXE              = 000000C0
PSC$M_GBL              = 00000010
PSC$M_LCL              = FFFFFFEF
PSC$M_LIB              = 00000002
PSC$M_LONG             = 00004800
PSC$M_NOEXE            = FFFFFFBF
PSC$M_NOPIC            = FFFFFFFE
PSC$M_NORD             = FFFFFF7F
PSC$M_NOSHR            = FFFFFFDF
PSC$M_NOVEC            = FFFFFDFF
PSC$M_NOWRT            = FFFFFEFF
PSC$M_OVR              = 00000004
PSC$M_PAGE             = 00006400
PSC$M_PIC              = 00000001
PSC$M_QUAD             = 00004C00
PSC$M_RD               = 00000080
PSC$M_REL              = 00000008
PSC$M_SHR              = 00000020
PSC$M_USR              = FFFFFFFD
```

```
PSC$M_VEC                       = 00000200
PSC$M_WORD                      = 00004400
PSC$M_WRT                       = 00000180
PSC$S_ALIGNMENT                 = 00000004
PSC$V_ALIGNFLG                  = 0000000E
PSC$V_ALIGNMENT                 = 0000000A
PSC$V_EXE                       = 00000006
PSC$V_GBL                       = 00000004
PSC$V_LIB                       = 00000001
PSC$V_OVR                       = 00000002
PSC$V_PIC                       = 00000000
PSC$V_RD                        = 00000007
PSC$V_REL                       = 00000003
PSC$V_SHR                       = 00000005
PSC$V_VEC                       = 00000009
PSC$V_WRT                       = 00000008
PSC$W_FLAG                        00000009
PSC$W_OPTIONS                     0000000D
PSECT$BLANK                       ********    X   03
PSECT$MAIN                        ********    X   03
RDX$V_BINARY                    = 00000000
RDX$V_DECIMAL                   = 00000002
RDX$V_DOUBLE                    = 00000005
RDX$V_FLOAT                     = 00000004
RDX$V_GFLOAT                    = 00000006
RDX$V_HEX                       = 00000003
RDX$V_HFLOAT                    = 00000007
RDX$V_OCTAL                     = 00000001
REG$_PC                         = 0000000F
SEMI                            = 0000003B
STB$K_PG_MISS                   = 0000000A
STS$M_SEVERITY                  = 00000007
STS$V_INHIB_MSG                 = 0000001C
SUM$CLOSE                         ********    X   03
SYM$B_NAME                        00000004
SYM$B_SEG                         0000000C
SYM$B_TOKEN                       0000000B
SYM$K_BLKSIZ                      0000000D
SYM$K_MAXLEN                    = 0000001F
SYM$K_TWOCOL                    = 00000010
SYM$L_LINK                        00000000
SYM$L_VAL                         00000005
SYM$M_ABS                       = 00000010
SYM$M_ASN                       = 00000100
SYM$M_CRFO                      = 00002000
SYM$M_DEBUG                     = 00000020
SYM$M_DEF                       = 00000001
SYM$M_DELMAC                    = 00000200
SYM$M_EPT                       = 00000200
SYM$M_EXTRN                     = 00000008
SYM$M_GLOBL                     = 00000004
SYM$M_LOCAL                     = 00000040
SYM$M_ODBG                      = 00000400
SYM$M_REF                       = 00000080
SYM$M_RELPSECT                  = 00000800
SYM$M_SUPR                      = 00004000
SYM$M_WEAK                      = 00000002
SYM$M_XCRF                      = 00001000
SYM$V_ABS                       = 00000004
SYM$V_ASN                       = 00000008
SYM$V_CRFO                      = 0000000D
SYM$V_DEBUG                     = 00000005
SYM$V_DEF                       = 00000000
SYM$V_DELMAC                    = 00000009
SYM$V_EPT                       = 00000009
SYM$V_EXTRN                     = 00000003
SYM$V_GLOBL                     = 00000002
SYM$V_LOCAL                     = 00000006
SYM$V_ODBG                      = 0000000A
SYM$V_REF                       = 00000007
SYM$V_RELPSECT                  = 0000000B
SYM$V_SUPR                      = 0000000E
SYM$V_WEAK                      = 00000001
SYM$V_XCRF                      = 0000000C
SYM$W_FLAG                        00000009
SYS$ASCTIM                        ********    GX  03
SYS$CLOSE                         ********    GX  03
SYS$CONNECT                       ********    GX  03
SYS$CREATE                        ********    GX  03
SYS$DISCONNECT                    ********    GX  03
SYS$EXIT                          ********    GX  03
TAB                             = 00000009
X1                              = 00000033
X2                              = 00080000
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK .	00000000 (0.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
ABS	00000177 (375.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
MAC$RO_CODE_COM	000003DA (986.)	03 (3.)	NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:01.55
Command processing	103	00:00:00.38	00:00:04.06
Pass 1	286	00:00:05.88	00:00:22.84
Symbol table sort	0	00:00:00.87	00:00:03.25
Pass 2	116	00:00:01.41	00:00:06.38
Symbol table output	50	00:00:00.19	00:00:01.56
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	587	00:00:08.83	00:00:39.66

The working set limit was 1350 pages.
52814 bytes (104 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 918 non-local and 22 local symbols.
502 source lines were read in Pass 1, producing 26 object records in Pass 2.
24 pages of virtual memory were used to define 22 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[MACRO.OBJ]MACRO.MLB;1	9
_255DUA28:[SYSLIB]STARLET.MLB;2	15
TOTALS (all libraries)	24

1017 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

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