
MMM MMM AAAAAAAAA CCCCCCCCCCC RRRRRRRRRRR 000000000
MMM MMM AAAAAAAAA CCCCCCCCCCC RRRRRRRRRRR 000000000
MMM MMM AAAAAAAAA CCCCCCCCCCC RRRRRRRRRRR 000000000
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 000000000
MMM MMM AAA AAA CCCCCCCCCCC RRR RRR 000000000
MMM MMM AAA AAA CCCCCCCCCCC RRR RRR 000000000
MMM MMM AAA AAA CCCCCCCCCCC RRR RRR 000000000

FILEID**DATA

MA
VO

```
DDDDDDDD      AAAAAA    TTTTTTTTTT    AAAAAA
DDDDDDDD      AAAAAA    TTTTTTTTTT    AAAAAA
DD      DD   AA    AA       TT      AA    AA
DD      DD   AA    AA       TT      AA    AA
DD      DD   AA    AA       TT      AA    AA
DD      DD   AA    AA       TT      AA    AA
DD      DD   AA    AA       TT      AA    AA
DD      DD   AA    AA       TT      AA    AA
DD      DD   AAAAAAAAAA     TT      AAAAAAAAAA
DD      DD   AAAAAAAAAA     TT      AAAAAAAAAA
DD      DD   AA    AA       TT      AA    AA      ....
DD      DD   AA    AA       TT      AA    AA      ....
DDDDDDDD     AA    AA       TT      AA    AA      ....
DDDDDDDD     AA    AA       TT      AA    AA      ....


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
```

```
0000     1          .TITLE  MAC$DATA        STORAGE ALLOCATION FOR VAX NATIVE ASSEMBLER
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: 30-AUG-78
0000    40  ;
0000    41  ; MODIFIED BY:
0000    42  ;
0000    43  ;       V03-03  MTR0020         Mike Rhodes     07-Jul-1982
0000    44  ;               Add MAC$GL_DSLISF disabled option flag cell to
0000    45  ;               allow for controlled overriding of macro directives
0000    46  ;               from the command level.
0000    47  ;
0000    48  ;       V03-02  MTR0018         Mike Rhodes     07-Jun-1982
0000    49  ;               Add MAC$GL_FNLSTS final status data cell which
0000    50  ;               contains the final exit status for MACRO, when
0000    51  ;               multiple assemblies are to be done.  This cell
0000    52  ;               holds the most severe status of all assemblies.
0000    53  ;
0000    54  ;       V03-01  MTR0014         Mike Rhodes     16-Apr-1982
0000    55  ;               Fix data declaration for MAC$GL_CVTADDR, which
0000    56  ;               caused occassional access violations.
0000    57  ;
```

```
0000     58 ;        V02.23  PCG0008          Peter George    26-Aug-1981
0000     59 ;                Fix the data allocation for MAC$GO_VALUE0.
0000     60 ;                Add MAC$GL_CVTADDR.
0000     61 ;
0000     62 ;        V02.22  CNH0045          Chris Hume      1-Dec-1980
0000     63 ;                Increased size of parser value stack.  Stack overflow remains
0000     64 ;                undetected.  Also advanced displayed IDENT to 2.46.
0000     65 ;                (DATA1.MAR 02.26, MAIN.MAR 02.46)
0000     66 ;
0000     67 ;        V02.21  HJ0002           Herb Jacobs     18-Aug-1980
0000     68 ;                Make $DEF's global using standard $GBLINI macro.
0000     69 ;
0000     70 ;        V01.20  RN0023           R. Newland      3-Nov-1979
0000     71 ;                New message codes to get error messages from system
0000     72 ;                message file.
0000     73 ;
0000     74 ;        V01.19  RN0022           R. Newland      31-Oct-1979
0000     75 ;                Translate SYS$LP_LINES to set lines/page
0000     76 ;
0000     77 ;        V01.19  RN0014           R. Newland      12-Oct-1979
0000     78 ;                Support for G_floating, H_floating and Octaword data types
0000     79 ;
0000     80 ;        V01.18  RN0011           R. Newland      11-Sep-1979
0000     81 ;                New librarian support
0000     82 ;
0000     83 ;        V01.17  RN0008           R. Newland      29-Aug-1979
0000     84 ;                31 character symbols
0000     85 ;
0000     86 ;        V01.16  RN0005           R. Newland      10-Aug-1979
0000     87 ;                Symbolically defined maximum argument size
0000     88 ;        V01.15  RN0002           R. Newland      01-Feb-1979
0000     89 ;                Changes for Source Update Merge
0000     90 ;        V01.17  RN0007           R. Newland      22-Aug-1979
0000     91 ;                Fix character table error for { \ } ~ and DEL.
0000     92 ;
0000     93 ;        V01.14  008              B. Schreiber              22-JAN-1979
0000     94 ;                Better bookkeeping of allocated pages.
0000     95 ;--
```

```
0000     97          .SBTTL  DECLARATIONS
0000     98 ;
0000     99 ; INCLUDE FILES:
0000    100 ;
0000    101
0000    102 ;
0000    103 ; MACROS:
0000    104 ;
0000    105
0000    106          $MAC_GRAMMARDEF          ;DEFINE VAX-11 MACRO TERMINAL GRAMMAR SYMBOLS
0000    107          $MAC_GENVALDEF           ;DEFINE VAX-11 MACRO COMMON SYMBOLS
0000    108          $MAC_INPBLKDEF           ;DEFINE INPUT BLOCK OFFSETS
003C    109          $MAC_SYMBLKDEF           ;SYMBOL BLOCK DEFINTIONS
0000    110          $FABDEF                  ;DEFINE FAB OFFSETS
0000    111          $NAMDEF                  ;DEFINE NAME BLOCK OFFSETS
0000    112          DEFSUMCBL                        ; Define SUM control block symbols
0000    113          $LBRDEF                          ; Define LBR offsets
0000    114
0000    115 ;
0000    116 ; EQUATED SYMBOLS:
0000    117 ;
0000    118
0000    119          $GBLINI GLOBAL          ;FORCE DATA ALLOCATION TO BE GLOBAL
0000    120
0000    121 ;
0000    122 ; OWN STORAGE:
0000    123 ;
0000    124
```

```
                     0000   126          .SBTTL  CHARACTER TOKEN TABLE
                     0000   127
                     0000   128 ;++
                     0000   129 ; FUNCTIONAL DESCRIPTION:
                     0000   130 ;
                     0000   131 ;       THE CHARACTER TABLE ('MAC$AL_CHRTAB') IS USED TO DETERMINE WHICH
                     0000   132 ;       TOKENS CAN POSSIBLY BE STARTED WITH A GIVEN CHARACTER.  THE
                     0000   133 ;       PROCEDURE IS TO OBTAIN THE FIRST CHARACTER OF THE TOKEN AND
                     0000   134 ;       GET THE TABLE ENTRY CORRESPONDING TO THE ASCII VALUE OF THE
                     0000   135 ;       CHARACTER.  IF THE CHARACTER ITSELF IS A TOKEN (LIKE DDPLUS)
                     0000   136 ;       THEN THE HIGH ORDER BIT WILL BE SET IN THE TABLE ENTRY.  IF
                     0000   137 ;       THE HIGH ORDER BIT IS NOT SET IT IS THE NAME OF A ROUTINE TO
                     0000   138 ;       CALL TO DECIDE THE TOKEN TYPE.  THIS ROUTINE MAY SCAN FURTHER
                     0000   139 ;       AS IN THE CASE OF A SYMBOL, OR IT MAY SIMPLY LOOK AHEAD TO
                     0000   140 ;       RESOLVE AN AMBIGUITY.  IN ANY CASE, THE ROUTINE INVOLVED WILL
                     0000   141 ;       RETURN THE TOKEN CLASS IN R8, THE ASSOCIATED VALUE (IF ANY)
                     0000   142 ;       IN MAC$VALUE, AND THE CHARACTER POINTER WILL BE UPDATED PAST
                     0000   143 ;       THE TOKEN SCANNED.
                     0000   144 ;
                     0000   145 ;       AN ADDITIONAL TABLE, MAC$AB_CMSK_TAB IS ALSO GENERATED.  THIS
                     0000   146 ;       IS A BYTE-ORIENTED TABLE, CONTAINING ONE BYTE FOR EACH CHARACTER.
                     0000   147 ;       THE VALUES ARE SET FROM THE FLAGS ARGUMENT AND ARE USED IN
                     0000   148 ;       SCANC/SPANC INSTRUCTIONS TO LOOK FOR A PARTICULAR TYPE OF CHARACTER
                     0000   149 ;
                     0000   150 ;
                     0000   151 ;
                     0000   152 ;
                     0000   153 ;--
                     0000   154
           80000000  0000   155          SPECIAL =       ^X80000000              ;FLAG THAT CHAR IS SPECIAL
                     0000   156
                     0000   157          .MACRO  $CHR_TABENTRY   VAL=0, SPF=0, FLAGS=0
                     0000   158          .LONG   SPF+VAL
                     0000   159          .PSECT  MAC$CHR_FLG_TAB,NOWRT,NOEXE,GBL,LONG
                     0000   160          .BYTE   FLAGS
                     0000   161          .PSECT  MAC$CHRTAB,NOWRT,NOEXE,GBL,LONG
                     0000   162          .ENDM   $CHR_TABENTRY
                     0000   163
                     0000   164          .PSECT  MAC$CHR_FLG_TAB,NOWRT,NOEXE,GBL,LONG
                     0000   165
                     0000   166 MAC$AB_CMSK_TAB::                          ;FLAG BITS FOR CHARACTERS
                     0000   167
                     0000   168          .PSECT  MAC$CHRTAB,NOWRT,NOEXE,GBL,LONG
                     0000   169
                     0000   170 MAC$AL_CHRTAB::                           ;CHARACTER TOKENS
                     0000   171
                     0000   172          $CHR_TABENTRY   0,,CHR$M_SPA_MSK!CHR$M_SYM_DLM  ;IGNORE NULLS
                     0004   173          .REPT   8
                     0004   174          $CHR_TABENTRY   MAC$CHRERR,,CHR$M_SYM_DLM ;0-8 ARE ILLEGAL CHARACTERS
                     0004   175          .ENDR
                     0024   176          $CHR_TABENTRY   0,,CHR$M_SPA_MSK!CHR$M_SYM_DLM  ;IGNORE TAB
                     0028   177          $CHR_TABENTRY   0,,CHR$M_SPA_MSK!CHR$M_SYM_DLM  ;IGNORE LINE FEED
                     002C   178          $CHR_TABENTRY   MAC$CHRERR,,CHR$M_SYM_DLM ;VERTICAL TAB IS ERROR
                     0030   179          $CHR_TABENTRY   0,,CHR$M_SPA_MSK!CHR$M_SYM_DLM  ;IGNORE FORM FEED
                     0034   180          $CHR_TABENTRY   DEOL,SPECIAL,CHR$M_COMMA_CR!CHR$M_SYM_DLM ;CR IS END OF LINE
                     0038   181          .REPT   18.
                     0038   182          $CHR_TABENTRY   MAC$CHRERR,,CHR$M_SYM_DLM ;CTRL-N TO CTRL-SHIFT-O ARE ERRORS
```

```
0038   183          .ENDR
0080   184          $CHR_TABENTRY   0,,CHR$M_SPA_MSK!CHR$M_SYM_DLM  ;IGNORE SPACE
0084   185          $CHR_TABENTRY   DOR,SPECIAL,CHR$M_SYM_DLM       ;(!)
0088   186          $CHR_TABENTRY   MAC$CHRERR,,CHR$M_SYM_DLM ;('') IS AN ERROR
008C   187          $CHR_TABENTRY   MAC$XPOUND,,CHR$M_SYM_DLM ;(#)
0090   188          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;($)
0094   189          $CHR_TABENTRY   MAC$CHRERR,,CHR$M_SYM_DLM       ;(%)
0098   190          $CHR_TABENTRY   DAND,SPECIAL,CHR$M_SYM_DLM      ;(&)
009C   191          $CHR_TABENTRY   MAC$CHRERR,,CHR$M_SYM_DLM       ;(')
00A0   192          $CHR_TABENTRY   DOPN,SPECIAL,CHR$M_SYM_DLM      ;(()
00A4   193          $CHR_TABENTRY   DCLS,SPECIAL,CHR$M_SYM_DLM      ;())
00A8   194          $CHR_TABENTRY   DTIMES,SPECIAL,CHR$M_SYM_DLM    ;(*)
00AC   195          $CHR_TABENTRY   DPLUS,SPECIAL,CHR$M_SYM_DLM     ;(+)
00B0   196          $CHR_TABENTRY   DCOMMA,SPECIAL,CHR$M_SYM_DLM!CHR$M_COMMA_CR ;(,)
00B4   197          $CHR_TABENTRY   DMINUS,SPECIAL,CHR$M_SYM_DLM    ;(-)
00B8   198          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(.)
00BC   199          $CHR_TABENTRY   DDIV,SPECIAL,CHR$M_SYM_DLM      ;(/)
00C0   200          .REPT   10.
00C0   201          $CHR_TABENTRY   MAC$NUMBER,,CHR$M_SYM_CH1!CHR$M_NUM_BER ;DIGITS
00C0   202          .ENDR
00E8   203          $CHR_TABENTRY   DCOLON,SPECIAL,CHR$M_SYM_DLM    ;(:)
00EC   204          $CHR_TABENTRY   MAC$CHRERR,,CHR$M_SYM_DLM       ;(;)
00F0   205          $CHR_TABENTRY   DANGOPN,SPECIAL,CHR$M_SYM_DLM   ;(<)
00F4   206          $CHR_TABENTRY   DEQ,SPECIAL,CHR$M_SYM_DLM       ;(=)
00F8   207          $CHR_TABENTRY   DANGCLS,SPECIAL,CHR$M_SYM_DLM   ;(>)
00FC   208          $CHR_TABENTRY   MAC$CHRERR,,CHR$M_SYM_DLM       ;(?)
0100   209          $CHR_TABENTRY   DAT,SPECIAL,CHR$M_SYM_DLM       ;(@)
0104   210          .REPT   6
0104   211          $CHR_TABENTRY   MAC$SYMNUM,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;LETTERS A-F CAN STA
0104   212          .ENDR
011C   213          $CHR_TABENTRY   MAC$XSYMBL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(G)
0120   214          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(H)
0124   215          $CHR_TABENTRY   MAC$XSYMBL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(I)
0128   216          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(J)
012C   217          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(K)
0130   218          $CHR_TABENTRY   MAC$XSYMBL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(L)
0134   219          .REPT   6
0134   220          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(M-R)
0134   221          .ENDR
014C   222          $CHR_TABENTRY   MAC$XSYMBL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(S)
0150   223          .REPT   3
0150   224          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(T-V)
0150   225          .ENDR
015C   226          $CHR_TABENTRY   MAC$XSYMBL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(W)
0160   227          .REPT   3
0160   228          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(X-Z)
0160   229          .ENDR
016C   230          $CHR_TABENTRY   DSQOPN,SPECIAL,CHR$M_SYM_DLM    ;([)
0170   231          $CHR_TABENTRY   DXOR,SPECIAL,CHR$M_SYM_DLM      ;(\)
0174   232          $CHR_TABENTRY   DSQCLS,SPECIAL,CHR$M_SYM_DLM    ;(])
0178   233          $CHR_TABENTRY   MAC$XUPARROW,,CHR$M_SYM_DLM     ;(^)
017C   234          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CH1!CHR$M_SYM_CHR ;(_)
0180   235          $CHR_TABENTRY   MAC$CHRERR,,CHR$M_SYM_DLM       ;(`)
0184   236          .REPT   6
0184   237          $CHR_TABENTRY   MAC$SYMNUM,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;LETTERS A-F CAN STA
0184   238          .ENDR
019C   239          $CHR_TABENTRY   MAC$XSYMBL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(G)
```

```
01A0   240          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(H)
01A4   241          $CHR_TABENTRY   MAC$XSYMBL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(I)
01A8   242          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(J)
01AC   243          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(K)
01B0   244          $CHR_TABENTRY   MAC$XSYMBL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(L)
01B4   245          .REPT   6
01B4   246          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(M-R)
01B4   247          .ENDR
01CC   248          $CHR_TABENTRY   MAC$XSYMBL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(S)
01D0   249          .REPT   3
01D0   250          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(T-V)
01D0   251          .ENDR
01DC   252          $CHR_TABENTRY   MAC$XSYMBL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(W)
01E0   253          .REPT   3
01E0   254          $CHR_TABENTRY   MAC$SYMBOL,,CHR$M_SYM_CHR!CHR$M_SYM_CH1 ;(X-Z)
01E0   255          .ENDR
01EC   256          .REPT   <255-^A/z/>             ;FILL OUT TABLE
01EC   257          $CHR_TABENTRY   MAC$CHRERR,,CHR$M_SPA_MSK!CHR$M_SYM_DLM
01EC   258          .ENDR
```

```
                                0400    260          .SBTTL  "XUPARROW" AND "XSYMBOL" CHARACTERS
                                0400    261
                                0400    262 ;++
                                0400    263 ;       THESE TABLES ARE USED TO SCAN RADIX CONTROL FUNCTIONS.
                                0400    264 ;       FUNCTIONS ARE EITHER "^"<CHAR> OR <CHAR>"^". "^" IS THE UPARROW
                                0400    265 ;       OR "HAT".
                                0400    266 ;       'MAC$AB_UPXTAB' POINTS TO A LIST OF CHARACTERS LEGAL AFTER A "^"
                                0400    267 ;       'MAC$AB_UPXTOKEN' POINTS TO A LIST OF CORRESPONDING TOKEN TYPES.
                                0400    268 ;
                                0400    269 ;       'MAC$AB_XUPTAB' POINTS TO A LIST OF CHARACTERS LEGAL BEFORE A "^"
                                0400    270 ;       'MAC$AB_XUPTOKEN' POINTS TO A CORRESPONDING LIST OF TOKEN TYPES.
                                0400    271 ;--
                                0400    272
                            00000000    273          .PSECT  MAC$RO_DATA,NOEXE,NOWRT,GBL,LONG
                                0000    274
                                0000    275 MAC$AB_UPXTAB::
64 63 62 61 58 4F 4D 46 44 43 42 41     0000    276          .ASCII  /ABCDFMOXabcdfmox/      ;^'X' CHARACTERS
                     78 6F 6D 66        000C
                        00000010        0010    277 LEN$K_UPXTAB==.-MAC$AB_UPXTAB
                                0010    278
                                0010    279          .ALIGN  LONG
                                0010    280 MAC$AB_UPXTOKEN::
            2A 27 29 28 26 25 24 23     0010    281          .BYTE   DUPA,DUPB,DUPC,DUPD,DUPF,DUPM,DUPO,DUPX
            2A 27 29 28 26 25 24 23     0018    282          .BYTE   DUPA,DUPB,DUPC,DUPD,DUPF,DUPM,DUPO,DUPX
                                0020    283
                                0020    284          .ALIGN  LONG
                                0020    285 MAC$AB_XUPTAB::
67 69 6C 77 62 73 47 49 4C 57 42 53     0020    286          .ASCII  /SBWLIGsbwlig/          ;'X'^ CHARACTERS
                        0000000C        002C    287 LEN$K_XUPTAB==.-MAC$AB_XUPTAB
                                002C    288
                                002C    289          .ALIGN  LONG
                                002C    290 MAC$AB_XUPTOKEN::
2C 2D 2E 30 2B 2F               002C    291          .BYTE   DSUP,DBUP,DWUP,DLUP,DIUP,DGUP
2C 2D 2E 30 2B 2F               0032    292          .BYTE   DSUP,DBUP,DWUP,DLUP,DIUP,DGUP
```

```
                     0038   294          .SBTTL  INITIALIZED DATA STORAGE DEFINITIONS
                     0038   295
                     0038   296 ;++
                     0038   297 ;       THIS SECTION DEFINES THE GLOBAL DATA STORAGE USED BY
                     0038   298 ;       THE VAX-11 MACRO ASSEMBLER THAT IS INITIALIZED AT
                     0038   299 ;       ASSEMBLY TIME.
                     0038   300 ;--
                     0038   301
                     0038   302          .ALIGN  LONG
                     0038   303
                     0038   304 MAC$GK_ZERO::
00000000             0038   305          .LONG   0                       ;A GUARANTEED ZERO WORD
                     003C   306 MAC$GK_ONE::                             ;A CONSTANT 1
00000001             003C   307          .LONG   1
                     0040   308 MAC$GK_INTSIZ::                         ;SIZE OF INTERMEDIATE BUFFER
00001400             0040   309          .LONG   INT$K_BUFSIZ+<3*4>      ;SIZE OF INT. BUFFER WE NEED
                     0044   310                                          ;TO ALLOCATE (2 LINK WORDS AND
                     0044   311                                          ;SIZE WORD)
                     0044   312 MAC$G_1_PAGE::
                     0044   313          $ARGLST 2,MAC$GK_1_PG_SIZ,-     ;ARG BLOCK TO ALLOCATE 1 PAGE
                     0044   314                  MAC$GL_BASEADDR
                     0050   315 MAC$G_2_PAGES::
                     0050   316          $ARGLST 2,MAC$GK_2_PG_SIZ,-     ;ARG BLOCK TO ALLOCATE 2 PAGES
                     0050   317                  MAC$GL_BASEADDR         ;AND STORE BASE ADDRESS HERE
                     005C   318 MAC$GK_1_PG_SIZ::
00000200             005C   319          .LONG   512                     ;SIZE OF ONE PAGE
                     0060   320 MAC$GK_2_PG_SIZ::                        ;SIZE OF TWO PAGES
00000400             0060   321          .LONG   1024                    ;...
                     0064   322 MAC$G_LSTBUFDES::                       ;DESCRIPTOR FOR FAO
00000418'            0064   323          .LONG   MAC$AB_LINE_END-MAC$AB_LST_END
00000002'            0068   324          .LONG   MAC$AB_LST_END
                     006C   325 MAC$AL_ATIM_DSC::                       ;STRING DESCRIPTOR TO GET ASCII TIME
00000014             006C   326          .LONG   20.                     ;TWENTY-THREE BYTE BUFFER
0000004A'            0070   327          .LONG   MAC$AB_ATIM_BUF         ;ADDRESS OF BUFFER
                     0074   328 MAC$AL_FTIM_DSC::                       ;STRING DESCRIPTOR FOR SUBTITLE LINE DATE
00000014             0074   329          .LONG   20.
000000CD'            0078   330          .LONG   MAC$AB_SBT_DATE
                     007C   331 MAC$GL_LIBTYPE::                        ; Library type = MACRO
00000002             007C   332          .LONG   LBR$C_TYP_MLB
                     0080   333 MAC$GL_LIBFUNC::                        ; Library function = READ
00000001             0080   334          .LONG   LBR$C_READ
                     0084   335 MAC$GQ_LINEBFDS::                       ; Descriptor for line buffer
00000032'000003E8    0084   336          .LONG   INP$K_BUFSIZ,MAC$AB_LINEBF
                     008C   337 MAC$GQ_LISTBFDS::                       ; Descriptor for listing buffer
00000002'000003E8    008C   338          .LONG   INP$K_BUFSIZ,MAC$AB_LST_END
```

```
                        .SBTTL  UNINITIALIZED DATA STORAGE DEFINITIONS

                ;++
                ;       THIS SECTION DEFINES THE GLOBAL DATA STORAGE USED BY THE
                ;       VAX-11 MACRO ASSEMBLER.
                ;--

00000000        .PSECT  MAC$RW_DATA,NOEXE,LONG

0000    $DEF    MAC$GL_FLAGS    .BLKL   2       ;GLOBAL ASSEMBLY FLAGS POINTED TO
0008                                            ; BY R11
0008    $DEF    MAC$GL_CLIADDR  .BLKL   1       ;CLI CALL BACK ADDRES
000C    $DEF    MAC$GL_CMDLIN   .BLKL   1       ;ADDRESS OF COMMAND LINE
0010    $DEF    MAC$GL_CMDLEN   .BLKL   1       ;LENGTH OF COMMAND LINE
0014    $DEF    MAC$GL_DIRFLG   .BLKL   1       ;FLAG WORD FOR DIRECTIVES
0018    $DEF    MAC$GL_ENLISF   .BLKL   1       ;FLAGS SET BY /SHOW/ENABLE IN COMMAND LINE
001C    $DEF    MAC$GL_DSLISF   .BLKL   1       ;FLAGS CLEARED BY /NOSHOW/DISABLE IN COMMAND
0020    $DEF    MAC$GL_INI_AP   .BLKL   1       ;INITIAL AP
0024    $DEF    MAC$GL_INI_FP   .BLKL   1       ;INITIAL FP
0028    $DEF    MAC$GL_INI_SP   .BLKL   1       ;INITIAL SP
002C    $DEF    MAC$GL_FNLSTS   .BLKL   1       ;FINAL EXIT STATUS
0030    ;***THE FOLLOWING 4 ITEMS MUST NOT BE SEPARATED
0030    ;
0030    $DEF    MAC$GB_MODE     .BLKB   1       ;PRIMARY MODE OF OPERAND
0031    $DEF    MAC$GB_IMODE    .BLKB   1       ;INDEXED MODE
0032    $DEF    MAC$GB_REG      .BLKB   1       ;REGISTER
0033    $DEF    MAC$GB_IREG     .BLKB   1       ;...
0034    ;
0034    ;***THE ABOVE 4 ITEMS MUST NOT BE SEPARATED
0034    $DEF    MAC$GL_IMP_BEG  .BLKL   0       ;BEGINNING OF IMPURE AREA
0034    $DEF    MAC$GL_ARGPTR   .BLKL   1       ;POINTER TO FREE SPOT ON PAGE
0038                                            ;DURING MACRO DEFINITION
0038    $DEF    MAC$GL_ASCCNT   .BLKL   1       ;CHARACTER COUNT FOR .ASCIX DIRECTIVES
003C    $DEF    MAC$GL_ABSFLAG  .BLKL   1       ;ABSOLUTE FLAG
0040    $DEF    MAC$GL_ASNPTR   .BLKL   1       ;POINTER TO SYM BLOCK FOR ASSIGNMENT EXPR
0044    $DEF    MAC$GL_BASEADDR .BLKL   1       ;RETURNS BASE ADDRESS FROM VM GET
0048    $DEF    MAC$GL_BLKPTR   .BLKL   1       ;POINTER TO INPUT BLOCK (GETARGS)
004C    $DEF    MAC$GL_CTLMSK   .BLKL   1       ;MASK OF FLAG BITS SET FROM CLI
0050    $DEF    MAC$GL_CRF_CNT  .BLKL   1       ;COUNT # TIMES CREF CALLED FOR SYMBOLDEF/REF
0054    $DEF    MAC$GL_CRF_FLG  .BLKL   1       ;FLAGS FOR DEF--WHAT TO CREF
0058    $DEF    MAC$GL_CRF_DCNT .BLKL   1       ;# DIRECTIVE DEF/REF CREF CALLS
005C    $DEF    MAC$GL_CRF_MCNT .BLKL   1       ;# MACRO DEF/REF CREF CALLS
0060    $DEF    MAC$GL_CRF_OCNT .BLKL   1       ;# OPCODE REF CALLS
0064    $DEF    MAC$GL_CRF_RCNT .BLKL   1       ;# REGISTER REF CALLS
0068    $DEF    MAC$GL_CRSYM    .BLKL   1       ;CREATED SYMBOL NUMBER
006C    $DEF    MAC$GL_CURINFDB .BLKL   1       ;POINTER TO CURRENT INPUT FDB
0070    $DEF    MAC$GL_DFPC_DSP .BLKL   1       ;0 OR INDEX FOR DEFAULT DISPLACEMENT
0074                                            ;FOR PC-RELATIVE W/NO "^"
0074    $DEF    MAC$GL_ERRCT    .BLKL   1       ;COUNT OF ERRORS ENCOUNTERED
0078    $DEF    MAC$GL_ERR_LIST .BLKL   2       ;QUEUE HEAD FOR ERROR LIST
0080    $DEF    MAC$AB_ETXBUF   .BLKB   512     ;ERROR TEXT BUFFER
0280    $DEF    MAC$GL_ETXLEN   .BLKL   1       ;LENGTH OF TEXT IN ETXBUF
0284    $DEF    MAC$GL_EXPOPVL1 .BLKL   1
0288    $DEF    MAC$GL_EXPOPVL2 .BLKL   1
028C    $DEF    MAC$GL_FINPTF   .BLKL   1       ;NON-ZERO WHEN FIRST INPUT FILE HAS BEEN PAR
0290    $DEF    MAC$GL_FREE_LST .BLKL   2       ;QUEUE HEAD FOR FREE PAGES
0298    $DEF    MAC$GL_HIGH_32  .BLKL   1       ;HIGH ORDER 32 BITS OF OPERAND
```

```
029C   397 $DEF     MAC$GQ_HIGH_64  .BLKQ   1          ;High order 64 bits of octaword operand
02A4   398 $DEF     MAC$GL_HSHVAL   .BLKL   1          ;HASH VALUE RETURNED FROM GETSYM
02A8   399 $DEF     MAC$AB_IDENT    .BLKB   SYM$K_MAXLEN+1 ;IDENT OF ASSEMBLY
02C8   400 $DEF     MAC$GL_IF_CNDPT .BLKL   1          ;POINTER TO IF CONDITION ROUTINE
02CC   401 $DEF     MAC$GL_IF_COUNT .BLKL   1          ;COUNT OF NESTED IF'S IN FALSE CONDITIONALS
02D0   402 $DEF     MAC$GL_IF_LEVEL .BLKL   1          ;'IF' LEVEL
02D4   403 $DEF     MAC$GL_IF_VALUE .BLKL   1          ;LOW BIT CLEAR IF CURRENT LEVEL IS TRUE
02D8   404 $DEF     MAC$GL_INFOCNT  .BLKL   1          ; Count of information messages
02DC   405 $DEF     MAC$GL_INTFRMPT .BLKL   1          ;POINTER TO CURRENT INT. FRAME BLOCK
02E0   406 $DEF     MAC$GL_INTWRNPT .BLKL   1          ;POINTER TO WARNING SPOT
02E4   407 $DEF     MAC$GL_INTPAGRQ .BLKL   1          ;# OF PAGES USED FOR INT. BUFFER
02E8   408 $DEF     MAC$GL_INTCNT   .BLKL   1          ;COUNT OF BYTES LEFT IN INTBUF
02EC   409 $DEF     MAC$GL_INTQUE   .BLKL   2          ;HEAD OF QUEUE FOR INT. BUFFERS
02F4   410 $DEF     MAC$GL_PRMINBL  .BLKB   INP$K_BLKSIZ ;PRIMARY INPUT BLOCK
0315   411           .ALIGN  LONG
0318   412 $DEF     MAC$GL_INPQUE   .BLKL   2          ;QUEUE HEAD TO INPUT FILE FDB LIST
0320   413 $DEF     MAC$GL_INPUTP   .BLKL   1          ;POINTER TO CURRENT INPUT BLOCK
0324   414 $DEF     MAC$GL_KEYMAC   .BLKL   1          ;
0328   415 $DEF     MAC$GL_KEYPTR   .BLKL   1          ;
032C   416 $DEF     MAC$GL_LINBAS   .BLKL   1          ;BASE LINE NUMBER OF CURRENT FILE
0330   417 $DEF     MAC$GL_LINENUM  .BLKL   1          ;SEQUENTIAL LINE # FOR ASSEMBLY
0334   418 $DEF     MAC$GL_LINELN   .BLKL   1          ;LENGTH OF CURRENT SOURCE LINE
0338   419 $DEF     MAC$GL_LINE_CNT .BLKL   1          ;NUMBER OF LINES REMAINING IN CURRENT PAGE
033C   420 $DEF     MAC$GL_LN_PAGE  .BLKL   1          ; # of text lines on page
0340   421 $DEF     MAC$GL_LINK_PTR .BLKL   1          ;POINTER TO LINKED, ALPHABETIZED SYMBOL TBL
0344   422 $DEF     MAC$GL_LIST_IT  .BLKL   1          ;LISTING FLAG
0348   423 $DEF     MAC$GL_LIST_LVL .BLKL   1          ;LISTING LEVEL
034C   424 $DEF     MAC$AB_LPBUF    .BLKB   16         ;BUFFER TO CREATE LINE/PAGE FOR CREF
035C   425 $DEF     MAC$GL_LPTPAG   .BLKL   1          ;CURRENT PAGE NUMBER
0360   426 $DEF     MAC$GL_LSB      .BLKL   1          ;LOCAL SYMBOL BLOCK NUMBER
0364   427 $DEF     MAC$GL_LSB_MAX  .BLKL   1          ;HIGHEST LSB ** MUST FOLLOW MAC$GL_LSB
0368   428 $DEF     MAC$GL_MACPTR   .BLKL   1          ;POINTER TO MACRO CURRENTLY BEING DEFINED
036C   429 $DEF     MAC$GL_MC_ARGCT .BLKL   1          ;MACRO ARGUMENT COUNT
0370   430 $DEF     MAC$GL_MCDEF    .BLKL   1          ;# MACROS DEFINED
0374   431 $DEF     MAC$GL_MCLVL    .BLKL   1          ;MACRO DEFINITION LEVEL
0378   432 $DEF     MAC$GL_MCPGRQ   .BLKL   1          ;# PAGES REQUIRED TO DEFINE MACROS
037C   433 $DEF     MAC$GL_MLB_CNT  .BLKL   1          ;# OF MLB'S WE KNOW ABOUT
0380   434 $DEF     MAC$GL_MLB_NDB  .BLKL   1          ;# INDEX BLOCKS USED FOR ALL MLB'S
0384   435 $DEF     MAC$GL_MLB_GET  .BLKL   1          ;# GETS TO DEFINE MACROS
0388   436 $DEF     MAC$GL_MLB_MDF  .BLKL   1          ;# MACROS DEFINED OUT OF ALL MLB'S
038C   437 $DEF     MAC$GL_MLB_QUE  .BLKL   2          ;MLB FDB QUEUE HEADER
0394   438 $DEF     MAC$GL_MLIN_LEN .BLKL   1          ;LENGTH OF MACRO LINE IN TMPBUF (P2)
0398   439 $DEF     MAC$GL_MOPNUM   .BLKL   1          ;NUMBER OF OPERANDS IN INSTR.
039C   440 $DEF     MAC$GL_MOPPTR   .BLKL   1          ;POINTER TO OPERAND DESCRIPTORS
03A0   441 $DEF     MAC$GL_OBJ_RCNT .BLKL   1          ;NO. OBJECT RECORDS WRITTEN
03A4   442 $DEF     MAC$GL_OPCPGPTR .BLKL   1          ;POINTER TO CURRENT OPDEF PAGES
03A8   443 $DEF     MAC$GL_OPCLSTPT .BLKL   1          ;POINTER TO OPDEF LIST
03AC   444 $DEF     MAC$GL_OPSIZE   .BLKL   1          ;NUMBER OF BYTES IN OPERAND
03B0   445 $DEF     MAC$GL_P2_LINE  .BLKL   1          ;LINE NUMBER IN PASS 2
03B4   446 $DEF     MAC$GL_PC       .BLKL   1          ;CURRENT PC
03B8   447 $DEF     MAC$GL_PRMSEG   .BLKL   1          ;SEGMENT OF EXPRESSION SYMBOL
03BC   448 $DEF     MAC$GL_PSC_BLKP .BLKL   1          ;POINTER TO FREE CORE FOR PSECT BLOCKS
03C0   449 $DEF     MAC$GL_PSC_LIST .BLKL   1          ;POINTER TO PSECT DEF. BLOCKS
03C4   450 $DEF     MAC$GL_PSC_MAX  .BLKL   1          ;HIGHEST PSECT # ALLOCATED
03C8   451 $DEF     MAC$AB_PSC_SBF  .BLKB   32         ;PSECT CONTEXT SAVE BUFFER
03E8   452 $DEF     MAC$AL_PSC_SLB  .BLKL   32         ;PSECT LSB CONTEXT SAVE BUFFER
0468   453 $DEF     MAC$GL_PSC_SBP  .BLKL   1          ;BYTE INDEX INTO CONTEXT SAVE BUFFER
```

```
                     046C   454 $DEF     MAC$GL_PSECT     .BLKL   1        ;NUMBER OF CURRENT PSECT
                     0470   455 $DEF     MAC$GL_PSECTPTR  .BLKL   1        ;POINTER TO CURRENT PSECT BLOCK
                     0474   456 $DEF     MAC$GB_RDXNDX    .BLKB   1        ;INDEX FOR CURRENT RADIX
                     0475   457 $DEF     MAC$GL_RECHDBUF  .BLKL   1        ;INPUT RECORD HEADER BUFFER
                     0479   458 $DEF     MAC$GL_RECTYP    .BLKL   1        ;RECORD TYPE BEING OUTPUT IN PASS 2
                     047D   459 $DEF     MAC$GL_SAVE_PC   .BLKL   1        ;SAVE PC FOR LISTING HERE
                     0481   460 $DEF     MAC$GL_STATUS    .BLKL   1        ;EXIT STATUS SET BY P2$END
                     0485   461 $DEF     MAC$GL_STOIMPTR  .BLKL   1        ;POINTER TO BEGINNING OF STORE IMMEDIATE COD
                     0489   462 $DEF     MAC$GL_SAVE_SP   .BLKL   1        ;SAVE STACK POINTER HERE
                     048D   463 $DEF     MAC$GL_SAV_BAS   .BLKL   1        ;SAVE LINE BASE
                     0491   464 $DEF     MAC$GL_SAV_LIN   .BLKL   1        ;SAVE LINE NO.
                     0495   465 $DEF     MAC$GL_SAV_PAG   .BLKL   1        ;AND PAGE NO. FOR CERTAIN GRAMMAR CONSTRUCTS
                     0499   466 $DEF     MAC$GL_SRC_LCNT  .BLKL   1        ;NO. SRC LINES READ IN PASS 1
                     049D   467 $DEF     MAC$GL_SRCPAG    .BLKL   1        ;SOURCE PAGE NUMBER
                     04A1   468 $DEF     MAC$GL_SYM_NLOC  .BLKL   1        ;NO. NON-LOCAL SYMBOLS
                     04A5   469 $DEF     MAC$GL_SYM_LOCL  .BLKL   1        ;NO. LOCAL SYMBOLS
                     04A9   470 $DEF     MAC$GL_SYM_PAGL  .BLKL   2        ;QUE OF ALLOCATED SYMBOL PAGES
                     04B1   471 $DEF     MAC$GL_SYMPGPTR  .BLKL   1        ;POINTER TO CURRENT PAGE OF SYMBOLS
                     04B5   472 $DEF     MAC$GL_SYMPGREQ  .BLKL   1        ;# OF GET_VM'S FOR SYMBOL TABLE
                     04B9   473 $DEF     MAC$GL_MLFPTR    .BLKL   1        ; Currect Macro library (MLF) pointer
                     04BD   474 $DEF     MAC$GL_TXTRFA    .BLKL   2        ; Librarian text RFA
           00000200  04C5   475         $$=512
           000003E8  04C5   476         .IIF GREATER <ARG$K_SIZE-$$>, $$=ARG$K_SIZE
                     04C5   477 $DEF     MAC$AB_TMPBUF    .BLKB   $$       ;TEMP BUFFER FOR ARGUMENTS, ETC.
                     08AD   478 $DEF     MAC$AB_TMPSYM    .BLKB   SYM$K_MAXLEN+1 ;TEMP SYMBOL NAME HOLDER
                     08CD   479 $DEF     MAC$AB_TMPSY1    .BLKB   SYM$K_MAXLEN+1 ;SECONDARY MAC$AB_TMPSYM
                     08ED   480 $DEF     MAC$AB_TITLE     .BLKB   SYM$K_MAXLEN+1 ;HOLDS STRING FROM .TITLE DIRECTIVE
                     090D   481 $DEF     MAC$GL_TTX_SIZ   .BLKL   1        ;LENGTH OF TITLE TEXT STRING
                     0911   482 $DEF     MAC$GL_WARNCT    .BLKL   1        ;COUNT OF WARNINGS ENCOUNTERED
                     0915   483 $DEF     MAC$GL_XFRADR    .BLKL   1        ;POINTER TO SYMBOL BLOCK FOR TRANSFER ADDRES
                     0919   484 $DEF     MAC$GW_LST_LINE  .BLKW   1        ; Listing line number
                     091B   485 $DEF     MAC$GW_LST_INST  .BLKW   1        ; Listing insert number
                     091D   486 $DEF     MAC$GT_SCB       .BLKB   SUM_K_BLN ; SUM control block
                     093A   487 $DEF     MAC$GL_CVTADDR   .BLKL   1        ; RTL floating pt. conversion routine addr
                     093E   488 $DEF     MAC$GL_IMP_END   .BLKL   0        ;END OF IMPURE AREA
           0000090A  093E   489         MAC$GK_IMP_SIZ==MAC$GL_IMP_END-MAC$GL_IMP_BEG
                     093E   490
                00000000    491         .PSECT  MAC$PARSE_DATA,NOEXE,LONG
                     0000   492
                     0000   493 $DEF     MAC$GL_ERRPT     .BLKL   1        ;POINTER TO LAST TOKEN SCANNED
                     0004   494 $DEF     MAC$GL_ERRPTX    .BLKL   1        ;POINTER TO CURRENT TOKEN SCANNED
                     0008   495 $DEF     MAC$GL_EXPPTR    .BLKL   1        ;POINTER TO EXPR START IN INT CODE
                     000C   496 $DEF     MAC$GL_EXPEND    .BLKL   1        ;POINTER TO EXPR END IN INT CODE
                     0010   497 $DEF     MAC$GL_NEXT      .BLKL   1        ;NEXT SYMBOL TO PARSE
                     0014   498 $DEF     MAC$AL_PSTACK    .BLKL   100.     ;PARSE STACK
                     01A4   499 $DEF     MAC$AL_VALSTACK  .BLKL   256.     ;PARSE VALUE STACK
                     05A4   500 $DEF     MAC$GL_VALUE     .BLKL   0        ;PARSER CURRENT VALUE
                     05A4   501 $DEF     MAC$GO_VALUEO    .BLKO   0        ; 128-bit value for number getters
                     05A4   502 $DEF     MAC$GQ_VALUEQ    .BLKQ   0        ;64-BIT VALUE FOR NUMBER-GETTERS
                     05A4   503 $DEF     MAC$GW_VAL1      .BLKW   0        ;FIRST WORD OF VALUE
                     05A4   504 $DEF     MAC$GB_VAL1      .BLKB   1        ;FIRST BYTE OF VALUE
                     05A5   505 $DEF     MAC$GB_VAL2      .BLKB   1        ;SECOND BYTE OF VALUE
                     05A6   506 $DEF     MAC$GW_VAL2      .BLKW   0        ;SECOND WORD OF VALUE
                     05A6   507 $DEF     MAC$GB_VAL3      .BLKB   1        ;THIRD BYTE OF VALUE
                     05A7   508 $DEF     MAC$GB_VAL4      .BLKB   1        ;FOURTH BYTE OF VALUE
                     05A8   509 $DEF     MAC$GL_VAL3      .BLKL   1        ;THIRD AND FOURTH WORDS OF VALUE
                     05AC   510 $DEF     MAC$GQ_VAL2      .BLKQ   1        ; Second quadword of octaword value
```

```
                05B4   511 $DEF     MAC$GL_VNEXT    .BLKL   1       ;VALUE DURING LOOKAHEAD
                05B8   512
            00000000   513         .PSECT  MAC$LISTING_BUF,NOEXE,LONG
                0000   514
            00000002   0000   515         .BLKB   2                       ;LISTING BUFFER OVERFLOW PROTECTION
                0002   516 $DEF     MAC$AB_LST_END  .BLKB   16      ;END OF CODE LISTING BUFFER
                0012   517 $DEF     MAC$AB_LST_OP2  .BLKB   14      ;START OF SECOND OPERAND FIELD
                0020   518 $DEF     MAC$AB_LST_OP1  .BLKB   5       ;START OF FIRST  OPERAND FIELD
                0025   519 $DEF     MAC$AB_LST_OPR  .BLKB   6       ;START OF OPCODE FIELD
                002B   520 $DEF     MAC$AB_SEQ_NUM  .BLKB   7       ;SOURCE LINE SEQUENCE NUMBER
                0032   521 $DEF     MAC$AB_LST_LIN  .BLKB   0       ;BEGINNING OF CODE LISTING BUFFER
            00000030   0032   522         MAC$K_LIST_SIZE==.-MAC$AB_LST_END       ;SIZE OF CODE LISTING BUFFER
            00000012   0032   523         MAC$AB_LST_AUDT = MAC$AB_LST_END+AUD$K_SIZE ; Start of audit trail
                0032   524                                 ; (THIS BUFFER GOES BACKWARDS!)
                0032   525 $DEF     MAC$AB_LINEBF   .BLKB   INP$K_BUFSIZ    ;SOURCE LINE BUFFER
                041A   526 $DEF     MAC$AB_LINE_END .BLKB   0       ;END OF SOURCE LINE BUFFER
            0000041E   041A   527         .BLKL   1                       ;OVERFLOW PADDING
                041E   528 $DEF     MAC$GL_LIST_PTR .BLKL   1       ;POINTER INTO MAC$AB_LST_LIN
                0422   529 $DEF     MAC$GL_LINEPT   .BLKL   1       ;POINTER INTO LINEBF
                0426   530
            00000000   531         .PSECT  MAC$PAGE_HEADER,NOEXE,LONG
                0000   532
                0000   533 MAC$AB_HD_NEWPG::                           ;TO OUTPUT FORM-FEED ALSO
            0C  0000   534         .BYTE   FF                      ;WANT NEW PAGE
                0001   535 $DEF     MAC$AB_HD_TITLE .BLKB   SYM$K_MAXLEN+1 ; Chars for title and sub string
                0021   536 $DEF     MAC$AB_HD_TSTRG .BLKB   LST$K_TITLE_SIZ+1
                004A   537                                 ;BUFFER FOR TITLE SUB-STRING
                004A   538 $DEF     MAC$AB_ATIM_BUF .BLKB   22      ; 22 bytes for date/time
                0060   539 $DEF     MAC$AB_HD_VERSN .BLKB   28      ; 28 bytes to hold assembler version string
                007C   540 $DEF     MAC$AB_HD_PAGE  .BLKB   8       ;"PAGE nnnn"
                0084   541 $DEF     MAC$AB_HD_END   .BLKB   0       ;END OF PAGE HEADER BUFFER
            00000083   0084   542         MAC$K_HD_SIZE==.-MAC$AB_HD_TITLE        ;SIZE
                0084   543 $DEF     MAC$AB_SBT_IDNT .BLKB   SYM$K_MAXLEN+1 ; Ident from .IDENT
                00A4   544 $DEF     MAC$AB_SBT_SBTL .BLKB   LST$K_TITLE_SIZ+1 ;SPACE FOR SUBTTL LINE
                00CD   545 $DEF     MAC$AB_SBT_DATE .BLKB   22      ; Creation date of source file
                00E3   546 $DEF     MAC$AB_SBT_FILE .BLKB   32      ; Source file specification
                0103   547 $DEF     MAC$AB_SBT_PAGE .BLKB   5       ;SOURCE PAGE NUMBER
                0108   548 $DEF     MAC$AB_SBT_END  .BLKB   0       ;END OF SUBTITLE LINE
            00000084   0108   549         MAC$K_SBT_SIZ==.-MAC$AB_SBT_IDNT
                0108   550
            00000000   551         .PSECT  MAC$PRO_TIMES,NOEXE,LONG
                0000   552
                0000   553 $DEF     MAC$GQ_RNT_TOT  .BLKQ   1       ;TOTAL CPU TIME FOR RUN
                0008   554 $DEF     MAC$GQ_TIM_TOT  .BLKQ   1       ;TOTAL ELAPSED TIME FOR RUN
                0010   555 $DEF     MAC$GL_PFL_TOT  .BLKL   1       ;PAGE FAULTS FOR TOTAL RUN
                0014   556 $DEF     MAC$GQ_RNT_CRF  .BLKQ   1       ;TOTAL CPU TIME FOR CREF
                001C   557 $DEF     MAC$GQ_TIM_CRF  .BLKQ   1       ;TOTAL ELAPSED TIME FOR CREF
                0024   558 $DEF     MAC$GQ_PFL_CRF  .BLKL   1       ;PAGE FAULTS FOR CREF
                0028   559 $DEF     MAC$GQ_RNT_INI  .BLKQ   1       ;TOTAL CPU TIME FOR INITIALIZATION
                0030   560 $DEF     MAC$GQ_TIM_INI  .BLKQ   1       ;ELAPSED TIME FOR INITIALIZATION
                0038   561 $DEF     MAC$GL_PFL_INI  .BLKL   1       ;PAGE FAULTS FOR INITIALIZATION
                003C   562 $DEF     MAC$GQ_RNT_CMD  .BLKQ   1       ;CPU TIME FOR COMMAND PROCESSING
                0044   563 $DEF     MAC$GQ_TIM_CMD  .BLKQ   1       ;ELAPSED TIME FOR COMMAND PROCESSING
                004C   564 $DEF     MAC$GL_PFL_CMD  .BLKL   1       ;PAGE FAULTS IN COMMAND PROCESSING
                0050   565 $DEF     MAC$GQ_RNT_P1   .BLKQ   1       ;CPU TIME FOR PASS 1
                0058   566 $DEF     MAC$GQ_TIM_P1   .BLKQ   1       ;ELAPSED TIME FOR PASS 1
                0060   567 $DEF     MAC$GL_PFL_P1   .BLKL   1       ;PAGE FAULTS IN PASS 1
```

```
0064    568 $DEF    MAC$GQ_RNT_SRT  .BLKQ   1       ;CPU TIME FOR SYMBOL TABLE SORT
006C    569 $DEF    MAC$GQ_TIM_SRT  .BLKQ   1       ;ELAPSED TIME FOR SYMBOL TABLE SORT
0074    570 $DEF    MAC$GL_PFL_SRT  .BLKL   1       ;PAGE FAULTS IN SYMBOL SORT
0078    571 $DEF    MAC$GQ_RNT_P2   .BLKQ   1       ;CPU TIME FOR PASS 2
0080    572 $DEF    MAC$GQ_TIM_P2   .BLKQ   1       ;ELAPSED TIME FOR PASS 2
0088    573 $DEF    MAC$GL_PFL_P2   .BLKL   1       ;PAGE FAULTS IN PASS 2
008C    574 $DEF    MAC$GQ_RNT_SYO  .BLKQ   1       ;CPU TIME FOR SYMBOL TABLE OUTPUT
0094    575 $DEF    MAC$GQ_TIM_SYO  .BLKQ   1       ;ELAPSED TIME FOR SYMBOL TABLE OUTPUT
009C    576 $DEF    MAC$GL_PFL_SYO  .BLKL   1       ;PAGE FAULTS IN SYMBOL TABLE OUTPUT
00A0    577 $DEF    MAC$GQ_RNT_PSY  .BLKQ   1       ;CPU TIME FOR PSECT SYNOPSIS OUTPUT
00A8    578 $DEF    MAC$GQ_TIM_PSY  .BLKQ   1       ;ELAPSED TIME FOR PSECT SYNO. OUTPUT
00B0    579 $DEF    MAC$GL_PFL_PSY  .BLKL   1       ;PAGE FAULTS FOR PSECT SYNO. OUTPUT
00B4    580
00B4    581         .END
```

```
$$                      = 000003E8
$$T1                    = 00000002
ARG$K_SIZE              = 000003E8
AUD$K_SIZE              = 00000010
BIT...                  = 00000005
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
CR                      = 0000000D
DAND                    = 0000001D
DANGCLS                 = 00000016
DANGOPN                 = 00000015
DAT                     = 00000020
DBUP                    = 0000002B
DCLS                    = 00000018
DCOLON                  = 00000010
DCOMMA                  = 0000000F
DDIV                    = 0000001C
DEOL                    = 0000000B
DEQ                     = 00000011
DGUP                    = 0000002C
DINTEGER                = 00000022
DIUP                    = 0000002D
DLUP                    = 0000002E
DMASK                   = 00000032
DMINUS                  = 0000001A
DOPCODE                 = 0000000E
DOPN                    = 00000017
DOR                     = 0000001E
DPC                     = 00000012
DPLUS                   = 00000019
DPOUND                  = 00000021
DSQCLS                  = 00000014
DSQOPN                  = 00000013
DSUP                    = 0000002F
DTIMES                  = 0000001B
DUPA                    = 00000023
DUPB                    = 00000024
DUPC                    = 00000025
DUPD                    = 00000026
DUPF                    = 00000028
DUPM                    = 00000029
DUPO                    = 00000027
DUPX                    = 0000002A
DWUP                    = 0000003C
DXOR                    = 0000001F
ERR01                   = 00000001
ERR02                   = 00000002
ERR03                   = 00000003
ERR04                   = 00000004
ERR05                   = 00000005
ERR06                   = 00000006
ERR07                   = 00000007
ERR08                   = 00000008
ERR09                   = 00000009
FF                      = 0000000C
GOALSY                  = 0000000A
HASHSZ                  = 0000007F
HYPHEN                  = 0000002D
ID                      = 0000000C
INP$B_ARGCT             = 0000001C
INP$K_BLKSIZ            = 00000021
INP$K_BUFSIZ            = 000003E8
INP$K_IRPSIZ            = 0000003C
INP$L_ARGS              = 0000001D
INP$L_GETL              = 00000008
INP$L_IFLVL             = 0000000C
INP$L_IFVAL             = 00000010
INP$L_LINK              = 00000000
INP$L_NXTL              = 00000004
INP$L_PAGP              = 00000018
INP$L_RPTCNT            = 00000014
INT$K_BUFSIZ            = 000013F4
INT$K_BUFWRN            = 00001390
KADDRESS                = 00000037
KALIGN                  = 0000005A
KASCIC                  = 00000033
KASCID                  = 00000078
KASCII                  = 00000034
KASCIZ                  = 00000035
KBLKA                   = 0000003F
KBLKB                   = 00000040
KBLKD                   = 00000041
KBLKF                   = 00000042
KBLKG                   = 0000007E
KBLKH                   = 0000007F
KBLKL                   = 00000043
KBLKO                   = 00000080
KBLKQ                   = 00000044
KBLKW                   = 00000045
KBYTE                   = 00000038
KCROSS                  = 00000079
KDEBUG                  = 00000055
KDFLT                   = 0000007B
KDOUBLE                 = 00000039
KDSABL                  = 00000056
KENABL                  = 00000057
KEND                    = 00000076
KENDC                   = 0000004E
KENDM                   = 00000053
```

KENDR	= 0000004F	LBR$C_READ	= 00000001				
KENTRY	= 00000058	LBR$C_TYP_MLB	= 00000002				61
KERROR	= 00000071	LEN$K_UPXTAB	= 00000010	G			20
KEVEN	= 0000005B	LEN$K_XUPTAB	= 0000000C	G			
KEXTRN	= 0000005D	LST$K_BUFSIZ	= 00000086				64
KFIELD	= 0000003A	LST$K_L_P_PAGE	= 0C00003C				
KFLOAT	= 0000003B	LST$K_TIT[illegible] SIZ	= 00000028				
KGFLOAT	= 00000081	MAB$B_ARGNL	00000005				31
KGLOBL	= 0000005E	MAB$B_NAME	00000004				
KHFLOAT	= 00000082	MAB$K_BLKSIZ	0000000C				
KIDENT	= 0000006A	MAB$L_DVPTR	00000008				
KIF	= 00000046	MAB$L_LINK	00000000				
KIFF	= 00000048	MAB$W_DVLEN	00000006				41
KIFT	= 00000049	MAC$AB_ATIM_BUF	0000004A	RG	09		20
KIFTF	= 0000004A	MAC$AB_CMSK_TAB	00000000	RG	03		
KIIF	= 00000047	MAC$AB_ETXBUF	00000080	RG	06		
KINCLUDE	= 0000005F	MAC$AB_HD_END	00000084	RG	09		53
KIRP	= 0000004B	MAC$AB_HD_NEWPG	00000000	RG	09		72
KIRPC	= 0000004C	MAC$AB_HD_PAGE	0000007C	RG	09		
KLIBRARY	= 00000060	MAC$AB_HD_TITLE	00000001	RG	09		
KLINK	= 00000085	MAC$AB_HD_TSTRG	00000021	RG	09		
KLIST	= 00000061	MAC$AB_HD_VERSN	00000060	RG	09		45
KLONG	= 0000003C	MAC$AB_IDENT	000002A8	RG	06		71
KMACRO	= 00000050	MAC$AB_LINEBF	00000032	RG	08		66
KMCALL	= 00000051	MAC$AB_LINE_END	0000041A	RG	08		72
KMDELETE	= 00000054	MAC$AB_LPBUF	0000034C	RG	06		
KMEXIT	= 00000052	MAC$AB_LST_AUDT	= 00000012	R	08		
KNARG	= 00000063	MAC$AB_LST_END	00000002	RG	08		
KNCHR	= 00000064	MAC$AB_LST_LIN	00000032	RG	08		41
KNCROS	= 0000007A	MAC$AB_LST_OP1	00000020	RG	08		20
KNLIST	= 00000062	MAC$AB_LST_OP2	00000012	RG	08		32
KNTYPE	= 00000074	MAC$AB_LST_OPR	00000025	RG	08		21
KOCTA	= 00000083	MAC$AB_PSC_SBF	000003C8	RG	06		
KODD	= 0000005C	MAC$AB_SBT_DATE	000000CD	RG	09		
KOPDEF	= 00000075	MAC$AB_SBT_END	00000108	RG	09		
KPACKED	= 00000036	MAC$AB_SBT_FILE	000000E3	RG	09		70
KPAGE	= 00000065	MAC$AB_SBT_IDNT	00000084	RG	09		
KPRINT	= 00000072	MAC$AB_SBT_PAGE	00000103	RG	09		
KPSECT	= 00000066	MAC$AB_SBT_SBTL	000000A4	RG	09		
KQUAD	= 0000003D	MAC$AB_SEQ_NUM	0000002B	RG	08		
KREF1	= 0000006D	MAC$AB_TITLE	000008ED	RG	06		2B
KREF16	= 00000084	MAC$AB_TMPBUF	000004C5	RG	06		
KREF2	= 0000006E	MAC$AB_TMPSY1	000008CD	RG	06		
KREF4	= 0000006F	MAC$AB_TMPSYM	000008AD	RG	06		21
KREF8	= 00000070	MAC$AB_UPXTAB	00000000	RG	05		6F
KREPT	= 0000004D	MAC$AB_UPXTOKEN	00000010	RG	05		
KRESTORE	= 00000067	MAC$AB_XUPTAB	00000020	RG	05		
KSAVE	= 0000006B	MAC$AB_XUPTOKEN	0000002C	RG	05		2B
KSBTTL	= 0000006B	MAC$AL_ATIM_DSC	0000006C	RG	05		
KSGNB	= 0000007C	MAC$AL_CHRTAB	00000000	RG	04		
KSGNW	= 0000007D	MAC$AL_FTIM_DSC	00000074	RG	05		
KTITLE	= 00000069	MAC$AL_PSC_SLB	000003E8	RG	06		53
KVECTOR	= 00000059	MAC$AL_PSTACK	00000014	RG	07		31
KWARN	= 00000073	MAC$AL_VALSTACK	000001A4	RG	07		61
KWEAK	= 0000006C	MAC$CHRERR	********	X	04		
KWORD	= 0000003E	MAC$GB_IMODE	00000031	RG	06		2E
KXFER	= 00000077	MAC$GB_IREG	00000033	RG	06		74

MAC$GB_MODE		00000030	RG	06	MAC$GL_INFOCNT		000002D8	RG	06			
MAC$GB_RDXNDX		00000474	RG	06	MAC$GL_INI_AP		00000020	RG	06			
MAC$GB_REG		00000032	RG	06	MAC$GL_INI_FP		00000024	RG	06			31
MAC$GB_VAL1		000005A4	RG	07	MAC$GL_INI_SP		00000028	RG	06			38
MAC$GB_VAL2		000005A5	RG	07	MAC$GL_INPQUE		00000318	RG	06			20
MAC$GB_VAL3		000005A6	RG	07	MAC$GL_INPUTP		00000320	RG	06			2A
MAC$GB_VAL4		000005A7	RG	07	MAC$GL_INTCNT		000002E8	RG	06			3E
MAC$GK_1_PG_SIZ		0000005C	RG	05	MAC$GL_INTFRMPT		000002DC	RG	06			
MAC$GK_2_PG_SIZ		00000060	RG	05	MAC$GL_INTPAGRQ		000002E4	RG	06			
MAC$GK_IMP_SIZ	=	0000090A	G		MAC$GL_INTQUE		000002EC	RG	06			35
MAC$GK_INTSIZ		00000040	RG	05	MAC$GL_INTWRNPT		000002E0	RG	06			29
MAC$GK_ONE		0000003C	RG	05	MAC$GL_KEYMAC		00000324	RG	06			
MAC$GK_ZERO		00000038	RG	05	MAC$GL_KEYPTR		00000328	RG	06			
MAC$GL_ABSFLAG		0000003C	RG	06	MAC$GL_LIBFUNC		00000080	RG	05			
MAC$GL_ARGPTR		00000034	RG	06	MAC$GL_LIBTYPE		0000007C	RG	05			41
MAC$GL_ASCCNT		00000038	RG	06	MAC$GL_LINBAS		0000032C	RG	06			35
MAC$GL_ASNPTR		00000040	RG	06	MAC$GL_LINELN		00000334	RG	06			28
MAC$GL_BASEADDR		00000044	RG	06	MAC$GL_LINENUM		00000330	RG	06			
MAC$GL_BLKPTR		00000048	RG	06	MAC$GL_LINEPT		00000422	RG	08			
MAC$GL_CLIADDR		00000008	RG	06	MAC$GL_LINE_CNT		00000338	RG	06			
MAC$GL_CMDLEN		00000010	RG	06	MAC$GL_LINK_PTR		00000340	RG	06			
MAC$GL_CMDLIN		0000000C	RG	06	MAC$GL_LIST_IT		00000344	RG	06			
MAC$GL_CRF_CNT		00000050	RG	06	MAC$GL_LIST_LVL		00000348	RG	06			
MAC$GL_CRF_DCNT		00000058	RG	06	MAC$GL_LIST_PTR		0000041E	RG	08			
MAC$GL_CRF_FLG		00000054	RG	06	MAC$GL_LN_PAGE		0000033C	RG	06			
MAC$GL_CRF_MCNT		0000005C	RG	06	MAC$GL_LPTPAG		0000035C	RG	06			
MAC$GL_CRF_OCNT		00000060	RG	06	MAC$GL_LSB		00000360	RG	06			
MAC$GL_CRF_RCNT		00000064	RG	06	MAC$GL_LSB_MAX		00000364	RG	06			
MAC$GL_CRSVM		00000068	RG	06	MAC$GL_MACPTR		00000368	RG	06			20
MAC$GL_CTLMSK		0000004C	RG	06	MAC$GL_MCDEF		00000370	RG	06			6F
MAC$GL_CURINFDB		0000006C	RG	06	MAC$GL_MCLVL		00000374	RG	06			73
MAC$GL_CVTADDR		0000093A	RG	06	MAC$GL_MCPGRQ		00000378	RG	06			74
MAC$GL_DFPC_DSP		00000070	RG	06	MAC$GL_MC_ARGCT		0000036C	RG	06			73
MAC$GL_DIRFLG		00000014	RG	06	MAC$GL_MLB_CNT		0000037C	RG	06			
MAC$GL_DSLISF		0000001C	RG	06	MAC$GL_MLB_GET		00000384	RG	06			
MAC$GL_ENLISF		00000018	RG	06	MAC$GL_MLB_MDF		00000388	RG	06			
MAC$GL_ERRCT		00000074	RG	06	MAC$GL_MLB_NDB		00000380	RG	06			20
MAC$GL_ERRPT		00000000	RG	07	MAC$GL_MLB_QUE		0000038C	RG	06			72
MAC$GL_ERRPTX		00000004	RG	07	MAC$GL_MLFPTR		000004B9	RG	06			77
MAC$GL_ERR_LIST		00000078	RG	06	MAC$GL_MLIN_LEN		00000394	RG	06			6E
MAC$GL_ETXLEN		00000280	RG	06	MAC$GL_MOPNUM		00000398	RG	06			
MAC$GL_EXPEND		0000000C	RG	07	MAC$GL_MOPPTR		0000039C	RG	06			74
MAC$GL_EXPOPVL1		00000284	RG	06	MAC$GL_NEXT		00000010	RG	07			21
MAC$GL_EXPOPVL2		00000288	RG	06	MAC$GL_OBJ_RCNT		000003A0	RG	06			73
MAC$GL_EXPPTR		00000008	RG	07	MAC$GL_OPCLSTPT		000003A8	RG	06			
MAC$GL_FINPTF		0000028C	RG	06	MAC$GL_OPCPGPTR		000003A4	RG	06			
MAC$GL_FLAGS		00000000	RG	06	MAC$GL_OPSIZE		000003AC	RG	06			6F
MAC$GL_FNLSTS		0000002C	RG	06	MAC$GL_P2_LINE		000003B0	RG	06			69
MAC$GL_FREE_LST		00000290	RG	06	MAC$GL_PC		000003B4	RG	06			20
MAC$GL_HIGH_32		00000298	RG	06	MAC$GL_PFL_CMD		0000004C	RG	0A			
MAC$GL_HSHVAL		000002A4	RG	06	MAC$GL_PFL_INI		00000038	RG	0A			
MAC$GL_IF_CNDPT		000002C8	RG	06	MAC$GL_PFL_P1		00000060	RG	0A			
MAC$GL_IF_COUNT		000002CC	RG	06	MAC$GL_PFL_P2		00000088	RG	0A			79
MAC$GL_IF_LEVEL		000002D0	RG	06	MAC$GL_PFL_PSY		000000B0	RG	0A			67
MAC$GL_IF_VALUE		000002D4	RG	06	MAC$GL_PFL_SRT		00000074	RG	0A			72
MAC$GL_IMP_BEG		00000034	RG	06	MAC$GL_PFL_SYO		0000009C	RG	0A			
MAC$GL_IMP_END		0000093E	RG	06	MAC$GL_PFL_TOT		00000010	RG	0A			74

MAC$GL_PRMINBL	000002F4	RG	06	MAC$GW_LST_INST		0000091B	RG	06		65
MAC$GL_PRMSEG	000003B8	RG	06	MAC$GW_LST_LINE		00000919	RG	06		74
MAC$GL_PSC_BLKP	000003BC	RG	06	MAC$GW_VAL1		000005A4	RG	07		
MAC$GL_PSC_LIST	000003C0	RG	06	MAC$GW_VAL2		000005A6	RG	07		
MAC$GL_PSC_MAX	000003C4	RG	06	MAC$G_1_PAGE		00000044	RG	05		
MAC$GL_PSC_SBP	00000468	RG	06	MAC$G_2_PAGES		00000050	RG	05		20
MAC$GL_PSECT	0000046C	RG	06	MAC$G_LSTBUFDES		00000064	RG	05		65
MAC$GL_PSECTPTR	00000470	RG	06	MAC$K_HD_SIZE	=	00000083	G			6F
MAC$GL_RECHDBUF	00000475	RG	06	MAC$K_LIST_SIZE	=	00000030	G			63
MAC$GL_RECTYP	00000479	RG	06	MAC$K_SBT_SIZ	=	00000084	G			
MAC$GL_SAVE_PC	0000047D	RG	06	MAC$NUMBER		********	X	04		4C
MAC$GL_SAVE_SP	00000489	RG	06	MAC$SYMBOL		********	X	04		61
MAC$GL_SAV_BAS	0000048D	RG	06	MAC$SYMNUM		********	X	04		6C
MAC$GL_SAV_LIN	00000491	RG	06	MAC$XPOUND		********	X	04		
MAC$GL_SAV_PAG	00000495	RG	06	MAC$XSYMBL		********	X	04		
MAC$GL_SRCPAG	0000049D	RG	06	MAC$XUPARROW		********	X	04		
MAC$GL_SRC_LCNT	00000499	RG	06	MACTXT	=	0000000D				6F
MAC$GL_STATUS	00000481	RG	06	MAC_SUBSYS	=	0000007D				53
MAC$GL_STOIMPTR	00000485	RG	06	MNB$B_ARGCT		00000017				6E
MAC$GL_SYMPGPTR	000004B1	RG	06	MNB$B_NAME		00000004				6F
MAC$GL_SYMPGREQ	000004B5	RG	06	MNB$K_BLKSIZ		0000001C				
MAC$GL_SYM_LOCL	000004A5	RG	06	MNB$L_ARGP		00000018				20
MAC$GL_SYM_NLOC	000004A1	RG	06	MNB$L_CRSYMF		00000013				6E
MAC$GL_SYM_PAGL	000004A9	RG	06	MNB$L_LINK		00000000				
MAC$GL_TTX_SIZ	0000090D	RG	06	MNB$L_PAGC		0000000F				
MAC$GL_TXTRFA	000004BD	RG	06	MNB$L_PAGP		0000000B				
MAC$GL_VAL3	000005A8	RG	07	MNB$L_TXTP		00000005				61
MAC$GL_VALUE	000005A4	RG	07	MNB$W_FLAG		00000009				72
MAC$GL_VNEXT	000005B4	RG	07	MXB$K_BLKSIZ		00000008				20
MAC$GL_WARNCT	00000911	RG	06	MXB$L_LINK		00000000				20
MAC$GL_XFRADR	00000915	RG	06	MXB$L_PAGES		00000004				
MAC$GO_VALUE0	000005A4	RG	07	OBJ$K_BUFSIZ	=	00000200				53
MAC$GQ_HIGH_64	0000029C	RG	06	PSC$B_NAME		00000004				
MAC$GQ_LINEBFDS	00000084	RG	05	PSC$B_SEG		0000000C				
MAC$GQ_LISTBFDS	0000008C	RG	05	PSC$B_UNUSED		0000000B				
MAC$GQ_PFL_CRF	00000024	RG	0A	PSC$K_BLKSIZ		00000013				
MAC$GQ_RNT_CMD	0000003C	RG	0A	PSC$K_NO_OPTNS	=	0000000A				
MAC$GQ_RNT_CRF	00000014	RG	0A	PSC$L_CURLOC		0000000F				
MAC$GQ_RNT_INI	00000028	RG	0A	PSC$L_LINK		00000000				
MAC$GQ_RNT_P1	00000050	RG	0A	PSC$L_MAXLGTH		00000005				
MAC$GQ_RNT_P2	00000078	RG	0A	PSC$M_ABS	=	FFFFFFF7				
MAC$GQ_RNT_PSY	000000A0	RG	0A	PSC$M_ALIGNFLG	=	00004000				74
MAC$GQ_RNT_SRT	00000064	RG	0A	PSC$M_ALLOPTNS	=	000003FF				
MAC$GQ_RNT_SYO	0000008C	RG	0A	PSC$M_BYTE	=	00004000				
MAC$GQ_RNT_TOT	00000000	RG	0A	PSC$M_CON	=	FFFFFFFB				
MAC$GQ_TIM_CMD	00000044	RG	0A	PSC$M_DEFAULT	=	000001C8				6F
MAC$GQ_TIM_CRF	0000001C	RG	0A	PSC$M_EXE	=	000000C0				
MAC$GQ_TIM_INI	00000030	RG	0A	PSC$M_GBL	=	00000010				
MAC$GQ_TIM_P1	00000058	RG	0A	PSC$M_LCL	=	FFFFFFEF				
MAC$GQ_TIM_P2	00000080	RG	0A	PSC$M_LIB	=	00000002				
MAC$GQ_TIM_PSY	000000A8	RG	0A	PSC$M_LONG	=	00004800				
MAC$GQ_TIM_SRT	0000006C	RG	0A	PSC$M_NOEXE	=	FFFFFFBF				
MAC$GQ_TIM_SYO	00000094	RG	0A	PSC$M_NOPIC	=	FFFFFFFE				6C
MAC$GQ_TIM_TOT	00000008	RG	0A	PSC$M_NORD	=	FFFFFF7F				
MAC$GQ_VAL2	000005AC	RG	07	PSC$M_NOSHR	=	FFFFFFDF				
MAC$GQ_VALUE0	000005A4	RG	07	PSC$M_NOVEC	=	FFFFFDFF				
MAC$GT_SCB	0000091D	RG	06	PSC$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
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$_PC                         =  0000000F
RRREG                           =  00000031
SEMI                            =  0000003B
SIZ...                          =  00000001
SPECIAL                         =  80000000
STB$K_PG_MISS                   =  0000000A
SUM_B_FLAGS                        0000001C
SUM_K_BLN                          0000001D
SUM_L_ISDATA                       00000004
SUM_L_STS                          00000000
SUM_M_AUDIT                     =  00000001
SUM_M_AUDITNEW                  =  00000002
SUM_M_DELETE                    =  00000010
SUM_M_SRCUPD                    =  00000004
SUM_M_SUBCLSH                   =  00000008
SUM_Q_AUDDS                        00000008
SUM_Q_FILESP                       00000010
SUM_V_AUDIT                     =  00000000
SUM_V_AUDITNEW                  =  00000001
SUM_V_DELETE                    =  00000004
SUM_V_SRCUPD                    =  00000002
SUM_V_SUBCLSH                   =  00000003
SUM_W_INSERT_NO                    0000001A
SUM_W_LINE_NO                      00000018
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
TAB                             =  00000009
X1                              =  00000400
X2                              =  0000000F
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK .	00000000 (0.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
ABS	0000003C (60.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
MAC$CHR_FLG_TAB	00000100 (256.)	03 (3.)	NOPIC USR CON REL GBL NOSHR NOEXE RD NOWRT NOVEC LONG
MAC$CHRTAB	00000400 (1024.)	04 (4.)	NOPIC USR CON REL GBL NOSHR NOEXE RD NOWRT NOVEC LONG
MAC$RO_DATA	00000094 (148.)	05 (5.)	NOPIC USR CON REL GBL NOSHR NOEXE RD NOWRT NOVEC LONG
MAC$RW_DATA	0000093E (2366.)	06 (6.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
MAC$PARSE_DATA	000005B8 (1464.)	07 (7.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
MAC$LISTING_BUF	00000426 (1062.)	08 (8.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
MAC$PAGE_HEADER	00000108 (264.)	09 (9.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
MAC$PRO_TIMES	000000B4 (180.)	0A (10.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.07	00:00:00.93
Command processing	108	00:00:00.46	00:00:02.84
Pass 1	324	00:00:10.23	00:00:38.89
Symbol table sort	0	00:00:00.96	00:00:04.04
Pass 2	151	00:00:01.88	00:00:06.94
Symbol table output	65	00:00:00.31	00:00:00.82
Psect synopsis output	3	00:00:00.04	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	683	00:00:13.96	00:00:54.52

The working set limit was 1800 pages.
68561 bytes (134 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 1008 non-local and 0 local symbols.
581 source lines were read in Pass 1, producing 54 object records in Pass 2.
18 pages of virtual memory were used to define 17 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SHRLIB]SUM.MLB;1	3
_255DUA28:[MACRO.OBJ]MACRO.MLB;1	5
_255DUA28:[SYSLIB]STARLET.MLB;2	8
TOTALS (all libraries)	16

943 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:DATA/OBJ=OBJ$:DATA MSRC$:DATA/UPDATE=(ENH$:DATA)+LIB$:MACRO/LIB+SHRLIB$:SUM/LIB
```

DATA
LIS

DEFINE
LIS

FLOAT
LIS

ERRMSG
LIS

GETARG
LIS

DATA1
LIS

INPUT
LIS

ERROR
LIS

FINISH
LIS

GETCMD
LIS