


```

EEEEEEEEEE  RRRRRRRR  RRRRRRRR  000000  RRRRRRRR
EEEEEEEEEE  RRRRRRRR  RRRRRRRR  000000  RRRRRRRR
EE          RR        RR        RR        RR  RR        RR
EE          RR        RR        RR        RR  RR        RR
EE          RR        RR        RR        RR  RR        RR
EEEEEEEEEE  RRRRRRRR  RRRRRRRR  00        00  RRRRRRRR
EEEEEEEEEE  RRRRRRRR  RRRRRRRR  00        00  RRRRRRRR
EE          RR  RR    RR  RR    RR  RR  RR
EE          RR  RR    RR  RR    RR  RR  RR
EE          RR    RR  RR    RR    RR  RR    RR
EE          RR    RR  RR    RR    RR  RR    RR
EEEEEEEEEE  RR        RR        RR        RR  RR        RR
EEEEEEEEEE  RR        RR        RR        RR  RR        RR

```

```

LL          IIIIII  SSSSSSSS
LL          IIIIII  SSSSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SSSSSS
LL          II      SSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLL IIIIII  SSSSSSSS

```

(2)	60
(3)	94

DECLARATIONS
ERROR ACTION ROUTINES

```

0000 1      .TITLE  MAC$ERROR      ERROR HANDLER FOR MACRO-32
0000 2      .IDENT  'V04-000'
0000 3
0000 4
0000 5      *****
0000 6      *
0000 7      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      *  ALL RIGHTS RESERVED.
0000 10     *
0000 11     *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     *  TRANSFERRED.
0000 17     *
0000 18     *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     *  CORPORATION.
0000 21     *
0000 22     *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     *
0000 25     *
0000 26     *  *****
0000 27     *
0000 28     *
0000 29     *+
0000 30     * FACILITY:      VAX MACRO ASSEMBLER OBJECT LIBRARY
0000 31     *
0000 32     * ABSTRACT:
0000 33     *
0000 34     * The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000 35     * modules for input to the VAX-11 LINKER.
0000 36     *
0000 37     * ENVIRONMENT:  USER MODE
0000 38     *
0000 39     * AUTHOR: Benn Schreiber, CREATION DATE: 26-SEP-1978
0000 40     *
0000 41     * MODIFIED BY:
0000 42     *
0000 43     *      V03.02  RR0012      Rowland R. Bradley      25-May-1984
0000 44     *              Add support for LIB$FIND_FILE error logging.
0000 45     *
0000 46     *      V03.01  RR0011      Rowland R. Bradley      01-Feb-1984
0000 47     *              Modify FABERR to handle FAB's which have no NAM block.
0000 48     *
0000 49     *      V03.00  MTR0004     Mike Rhodes      15-Mar-1982
0000 50     *              Modify routine MAC$PUT_MSG to flag any "external errors"
0000 51     *              that is non-assembly errors issued by facilities such as
0000 52     *              RMS, LBR, SUM, etc., in the flags word bits FLG$V_EXTERR
0000 53     *              and FLG$V_EXTWRN. Fixes SPR #11-41651(A).
0000 54     *
0000 55     *      V01.04  RN0011      R. Newland      11-Sep-1979
0000 56     *              New librarian support
0000 57     *

```

MAC\$ERROR
V04-000

ERROR HANDLER FOR MACRO-32

D 7

16-SEP-1984 02:04:13 VAX/VMS Macro V04-00
5-SEP-1984 01:48:07 [MACRO.SRC]ERROR.MAR;1

Page 2
(1)

MAC
V04

0000 58 ;--

```

0000 60          .SBTTL  DECLARATIONS
0000 61          :
0000 62          : INCLUDE FILES:
0000 63          :
0000 64          :
0000 65          :
0000 66          : MACROS:
0000 67          :
0000 68          :
0000 69          $FABDEF          ;DEFINE FAB OFFSETS
0000 70          $SHRDEF          ;DEFINE SHARED MESSAGES
0000 71          $MAC_CTLFLGDEF   ;DEFINE CONTROL FLAGS
0000 72          $STSDEF          ;DEFINE STATUS WORD BITS AND STUFF
0000 73          $NAMDEF          ;DEFINE NAM OFFSETS
0000 74          $RABDEF          ;DEFINE RAB OFFSETS
0000 75          $MAC_MLFDEF      ; Define MLF offsets
0177 76          $RMSDEF          ; Define RMS return codes
0177 77          :
0177 78          :
0177 79          : OWN STORAGE
0177 80          :
00000000 81          .PSECT  MAC$RO_DATA,NOWRT,NOEXE,GBL,LONG
0000 82          :
0000 83          MAC$MACRO_NAME::  ;MY NAME
00000008'00000005' 0000 84          .LONG  20$-10$,10$
4F 52 43 41 4D 0008 85          10$:  .ASCII  \MACRO\
000D 86          20$:
000D 87          :
00000FFC 000D 88          ERR_MASK=^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
00030000 000D 89          CLI^SUBSYS=^X30000          ;***TEMP
00000010 000D 90          SRMSL_SAVE_PC=16          ;***TEMP--SAVED PC IN CALL FRAME
000D 91          :
00000000 92          .PSECT  MAC$RO_CODE_P4,NOWRT,GBL,LONG

```

```

0000 94      .SBTTL  ERROR ACTION ROUTINES
0000 95
0000 96      :++
0000 97      : Functional description:
0000 98      :
0000 99      :     This routine is called to report librarian open errors
0000 100     :
0000 101     : Inputs:
0000 102     :     4(AP) = Address of MLF
0000 103     :     R0   = LBR$ status code
0000 104     :
0000 105     :--
0000 106
0000 107     MAC$ERR_LBROPEN::
51  50  0C  10  OFFC 0000 108     .WORD  ERR_MASK
0000 109     EXTZV  #STSSV FAC NO,#STSS$ FAC_NO,R0,R1 ; Get facility number
0000 110     CMPL  R1,#<RMS$_NORMAL@-165 ; Is it an RMS error?
0000 111     BEQL  10$ ; Yes if EQL
0000 112     PUSHL #0 ; No FAO arguments
0000 113     PUSHL R0 ; Status code
0000 114     PUSHL #2 ; Length of message packet
0000 115     BRW  MAC$PUT_MSG ; Output message and return
0000 116     10$:
0000 117     MOVL  4(AP),R1 ; Get MLF address
0000 118     PUSHL R0 ; RMS error code
0000 119     PUSHL #0
0000 120     PUSHAB MLF$Q_FNAMDS(R1) ; File name descriptor
0000 121     PUSHL #1 ; Number of FAO arguments
0003109A 8F DD 0022 122     PUSHL #<SHR$_OPENIN!CLI_SUBSYS!STSS$ ERROR> ; Error code
0000 123     PUSHL #5 ; Length of message packet
0000 124     BRW  MAC$PUT_MSG ; Output message and return
0000 125
0000 126
0000 127
0000 128     :++
0000 129     : FUNCTIONAL DESCRIPTION:
0000 130     :
0000 131     :     THIS ROUTINE REPORTS AN ERROR. R0 CONTAINS THE ADDRESS OF
0000 132     :     A STRING DESCRIPTOR TO OUTPUT USING SHR$_TEXT.
0000 133     :
0000 134     :--
0000 135
0000 136     MAC$ERR_TEXT::
0000 137     .WORD  ERR_MASK
0000 138     PUSHL R0 ; STACK DESCRIPTOR ADDRESS
0000 139     PUSHL #1 ; ONE FAO ARGUMENT
00031132 8F DD 0033 140     PUSHL #<SHR$_TEXT!CLI_SUBSYS!STSS$ ERROR> ; MESSAGE CODE
0000 141     PUSHL #3 ; THREE ARGS IN PACKET
0000 142     BRW  MAC$PUT_MSG ; PUT MESSAGE AND RETURN
0000 143
0000 144     :++
0000 145     : FUNCTIONAL DESCRIPTION:
0000 146     :
0000 147     :     THIS ROUTINE IS CALLED WHEN THERE IS NO MEMORY AVAILABLE.
0000 148     :     THE ERROR IS REPORTED. IF PASS 1 IS IN PROGRESS, A JUMP
0000 149     :     TO MAC$ABORT_PASS1 IS EXECUTED. IF PASS 2 IS IN PROGRESS,
0000 150     :     ?

```

```

003E 151 :
003E 152 :--
003E 153 :
003E 154 MAC$ERR_NOMEM::
003E 155     CALLS #0,W^MAC$ERR_NOMEM_0 ;SO WE CAN RETURN CORRECTLY
FFBA' 31 0043 156     BRW   MAC$LAST_CHANCE ;PASS 2--NOTHING TO DO BUT DIE
0046 157 :
0046 158 MAC$ERR_NOMEM_0::
0046 159     .WORD  ERR_MASK ;ENTRY MASK
0048 160     PUSHL #0 ;NO FAO ARGS
004A 161     PUSHL R0 ;STACK ERROR CODE
004C 162     PUSHL #2 ;TOTAL SIZE OF PACKET
00F2 31 004E 163     BRW   MAC$PUT_MSG ;OUTPUT MESSAGE AND RETURN
0051 164 :
0051 165 :
0051 166 :++
0051 167 : FUNCTIONAL DESCRIPTION:
0051 168 :
0051 169 :     THIS ROUTINE IS CALLED TO REPORT A BAD KEYWORD VALUE ERROR.
0051 170 :     THE KEYWORD IN ERROR IS IN MAC$AB_TPSYM.
0051 171 :
0051 172 :--
0051 173 :
0051 174 MAC$ERR_KEY_WD::
0051 175     .WORD  ERR_MASK
50 00000000'EF 9E 0053 176     MOVAB  L^MAC$AB_TPSYM,R0 ;POINT TO THE KEYWORD IN ERROR
01 A0 9F 005A 177     PUSHB  1(R0) ;STACK ITS ADDRESS
7E 60 9A 005D 178     MOVZBL (R0),-(SP) ;STACK ITS LENGTH
6E 7F 0060 179     PUSHAQ 0(SP) ;STACK POINTER TO DESCRIPTOR
01 DD 0062 180     PUSHL  #1 ;NUMBER OF ARGUMENTS
00031112 8F DD 0064 181     PUSHL  #<SHR$_BADVALUE!CLI_SUBSYS!ST$K_ERROR> ;STACK MESSAGE CODE
03 DD 006A 182     PUSHL  #3 ;TOTAL SIZE OF PACKET
00D4 31 006C 183     BRW   MAC$PUT_MSG ;OUTPUT ERROR AND RETURN TO CALLER
006F 184 :
006F 185 :++
006F 186 : FUNCTIONAL DESCRIPTION:
006F 187 :
006F 188 :     THIS ROUTINE IS CALLED WHEN AN INTERNAL ERROR IS DETECTED.
006F 189 :     THE ERROR IS REPORTED, AND THE ASSEMBLY IS ABORTED.
006F 190 :
006F 191 :--
006F 192 :
006F 193 MAC$ERR_INTERN::
006F 194     .WORD  ERR_MASK
10 AD DD 0071 195     PUSHL  SRMSL_SAVE_PC(FP) ;STACK ADDRESS OF CALLER'S RETURN
01 DD 0074 196     PUSHL  #1 ;NUMBER OF ARGUMENTS
00031152 8F DD 0076 197     PUSHL  #<SHR$_BADLOGICPC!CLI_SUBSYS!ST$K_ERROR> ;ERROR CODE
03 DD 007C 198     PUSHL  #3 ;TOTAL SIZE OF PACKET
00C2 31 007E 199     BRW   MAC$PUT_MSG ;CALL PUTMSG AND RETURN
0081 200 :
0081 201 :
0081 202 :++
0081 203 : FUNCTIONAL DESCRIPTION:
0081 204 :
0081 205 :     THESE ROUTINES ARE CALLED FOR INPUT AND OUTPUT ERRORS.
0081 206 :
0081 207 :--

```



```

0081 208
0081 209 MAC$ERR_FIND::
0081 210 MAC$ERR_MODIFY::
0081 211 MAC$ERR_CONNECT::
0081 212 MAC$ERR_DISCON::
      OFFC 0081 213          .WORD  ERR MASK
14 10 0083 214          BSBB  RABERR
000311CA 0085 215          .LONG  SHR$_RMSERROR!CLI_SUBSYS!ST$K_ERROR
0089 216
0089 217 MAC$ERR_GET::
      OFFC 0089 218          .WORD  ERR MASK          ;ENTRY MASK
OC 10 008B 219          BSBB  RABERR          ;PROCESS ERROR ON A RAB
000310B2 008D 220          .LONG  SHR$_READERR!CLI_SUBSYS!ST$K_ERROR
0091 221
0091 222 MAC$ERR_PUT::
      OFFC 0091 223          .WORD  ERR MASK
04 10 0093 224          BSBB  RABERR          ;PROCESS RAB ERROR
000310D2 0095 225          .LONG  SHR$_WRITEERR!CLI_SUBSYS!ST$K_ERROR
0099 226
0099 227 RABERR: MOVL 4(AP),R1          ;GET THE RAB ADDRESS
50 04 AC DO 009D 228          MOVL  RAB$_FAB(R1),R0          ;GET THE FAB ADDRESS
50 3C A1 DO 00A1 229          MOVQ  RAB$_STS(R1),FAB$_STS(R0) ;PUT ERROR IN COMMON PLACE
08 A0 08 A1 7D 00A6 230          BxB  FILERR          ;PROCESS ERROR
      00A8 231
      00A8 232 :++
      00A8 233 : FUNCTIONAL DESCRIPTION:
      00A8 234 :
      00A8 235 : THESE ROUTINES ARE CALLED TO PROCESS ERRORS FINDING AND OPENING
      00A8 236 : INPUT OR OUTPUT FILES, AND ERRORS CLOSING OUTPUT FILES.
      00A8 237 :
      00A8 238 :--
      00A8 239
00A8 240 MAC$ERR_FIND_INP::          ; Lib$find_file Error Reporting
      OFFC 00A8 241          .WORD  ERR MASK
50 0C AC DO 00AA 242          MOVL  12(AP),R0          ;GET SECOND RMS ERROR STS
54 08 AC DO 00AE 243          MOVL  8(AP),R4          ;GET STV ADDR ADDRESS
51 04 AC DO 00B2 244          MOVL  4(AP),R1          ;GET RESULT ADDRESS
53 0000'C1 DO 00B6 245          MOVL  DSC$_A_POINTER(R1),R3          ;SET RESULTANT STRING ADDRESS
52 0000'C1 9A 00BB 246          MOVZBL DSC$_LENGTH(R1),R2          ;AND LENGTH
      0C BB 00C0 247          PUSHR  #^M<R2,R3>          ;PUSH DESCRIPTOR FOR FILE NAME
      54 DD 00C2 248          PUSHL  R4          ;RMS STV
      50 DD 00C4 249          PUSHL  R0          ;RMS STS
      08 AE 7F 00C6 250          PUSHAQ 8(SP)          ;ADDRESS OF NAME DESCRIPTOR
      01 DD 00C9 251          PUSHL  #1          ;NUMBER OF ARGUMENTS
0003109A 8F DD 00CB 252          PUSHL  #<SHR$_OPENIN!CLI_SUBSYS!ST$K_ERROR>
      00D1 253          ;MESSAGE CODE
      00D1 254          ;
      05 DD 00D1 255          PUSHL  #5          ;TOTAL SIZE OF PACKET
      00D3 256          ;PROCESS LIB$FIND_FILE ERROR WITH
      00D3 257          ; RMS ERROR.
      006D 31 00D3 258          BRW  MAC$PUT_MSG          ;CALL PUTMSG AND RETURN
00000000 00D6 259          .LONG
00DA 260
00DA 261 MAC$ERR_OPN_INP::
      OFFC 00DA 262          .WORD  ERR MASK
1C 10 00DC 263          BSBB  FABERR          ;PROCESS FAB ERROR
0003109A 00DE 264          .LONG  SHR$_OPENIN!CLI_SUBSYS!ST$K_ERROR

```

```

00E2 265
00E2 266 MAC$ERR_OPN OUT::
OFFC 00E2 267 .WORD ERR_MASK
14 10 00E4 268 BSBB FABERR ;PROCESS FAB ERROR
000310A2 00E6 269 .LONG SHRS_OPENOUT!CLI_SUBSYS!STSSK_ERROR
00EA 270
00EA 271 MAC$ERR_CLS INP::
OFFC 00EA 272 .WORD ERR_MASK
OC 10 00EC 273 BSBB FABERR
00031052 00EE 274 .LONG SHRS_CLOSEIN!CLI_SUBSYS!STSSK_ERROR
00F2 275
00F2 276 MAC$ERR_CLS OUT::
OFFC 00F2 277 .WORD ERR_MASK
04 10 00F4 278 BSBB FABERR
0003105A 00F6 279 .LONG SHRS_CLOSEOUT!CLI_SUBSYS!STSSK_ERROR
00FA 280
50 04 AC D0 00FA 281 FABERR: MOVL 4(AP),R0 ;GET THE FAB ADDRESS
51 28 A0 D0 00FE 282 FILERR: MOVL FAB$L_NAM(R0),R1 ;GET NAME BLOCK ADDRESS
14 13 0102 283 BEQL 5$ ;IN CASE NO NAM BLOCK ?
53 04 A1 D0 0104 284 MOVL NAM$L_RSA(R1),R3 ;SET RESULTANT STRING ADDRESS
52 03 A1 9A 0108 285 MOVZBL NAM$B_RSL(R1),R2 ;AND LENGTH
1C 12 010C 286 BNEQ 10$ ;IF NEQ RESULTANT NAME FORMED
53 0C A1 D0 010E 287 MOVL NAM$L_ESA(R1),R3 ;GET ADDRESS OF EXPANDED STRING
52 0B A1 9A 0112 288 MOVZBL NAM$B_ESL(R1),R2 ;AND ITS LENGTH
12 12 0116 289 BNEQ 10$ ;IF NEQ EXPANDED STRING FORMED
53 2C A0 D0 0118 290 5$: MOVL FAB$L_FNA(R0),R3 ;USE INPUT FILE NAME
52 34 A0 9A 011C 291 MOVZBL FAB$B_FNS(R0),R2 ;AND SIZE
08 12 0120 292 BNEQ 10$ ;IF NEQ GO USE IT
53 30 A0 D0 0122 293 MOVL FAB$L_DNA(R0),R3 ;TRY THE DEFAULT NAME NOW
52 35 A0 9A 0126 294 MOVZBL FAB$B_DNS(R0),R2 ;AND SIZE
51 9E D0 012A 295 10$: MOVL @ (SP)+,R1 ;GET THE ERROR CODE
OC BB 012D 296 PUSHR #*M<R2,R3> ;PUSH DESCRIPTOR FOR FILE NAME
7E 08 A0 7D 012F 297 MOVQ FAB$L_STS(R0),-(SP) ;PUSH RMS ERROR CODES
6E 827A 8F B1 0133 298 CMPW #<RMS$ EOF&^XFFFF>,(SP) ;WAS ERROR END OF FILE?
47 13 0138 299 BEQL ERROR_EXIT ;IF EQL YES--NO ERROR NEEDED
08 AE 7F 013A 300 PUSHAQ 8(SP) ;ADDRESS OF NAME DESCRIPTOR
01 DD 013D 301 PUSHL #1 ;NUMBER OF ARGUMENTS
51 DD 013F 302 PUSHL R1 ;MESSAGE CODE
05 DD 0141 303 PUSHL #5 ;TOTAL SIZE OF MESSAGE PACKET
0143 304
0143 305 ; NOW SETUP TO CALL $PUTMSG TO OUTPUT THE MESSAGE
0143 306
0143 307 MAC$PUT_MSG::
00000000'EF 04 AE D0 0143 308 MOVL 4(SP),L^MAC$GL_STATUS ;SAVE MOST CURRENT STATUS
5B 00000000'EF 9E 014B 309 MOVAB MAC$GL_FLAGS,R1 ;POINT TO THE GLOBAL ASSEM FLAGS
FFFFFFF8 8F CB 0152 310 BICL3 #^CSTSSM_SEVERITY,- ;COPY SEVERITY BITS FROM STATUS
52 00000000'GF 0158 311 G^MAC$GL_STATUS,R2 ;IS SEVERITY = WARNING?
0B 13 015E 312 BEQL 10$ ;BRANCH IF YES.
03 52 D1 0160 313 CMPL R2,#STSSK_INFO ;NO -- IS IT INFORMATION?
0A 13 0163 314 BEQL 20$ ;YES -- INFO DOESN'T COUNT
06 6B 30 E3 0165 315 BBCS #FLG$V_EXTErr,(R11),20$ ;ITS AN ERROR, SO FLAG IT.
04 11 0169 316 BRB 20$ ;WATCH OUT FOR SUCCESSIVE ERRORS.
00 6B 31 E3 016B 317 10$: BBCS #FLG$V_EXTWRN,(R11),20$ ;ITS A WARNING, FLAG IT.
00000000'EF 7F 016F 318 20$: PUSHAQ MAC$MACRO_NAME ;PUSH ADDRESS OF FACILITY NAME
7E D4 0175 319 CLRL -(SP) ;ZERO ADDRESS OF ACTION ROUTINE
08 AE 9F 0177 320 PUSHAQ 8(SP) ;PUSH ADDRESS OF MESSAGE BUFFER
00000000'GF 03 FB 017A 321 CALLS #3,G^SYSS$PUTMSG ;PRINT THE MESSAGE

```

MAC\$ERROR
V04-000

ERROR HANDLER FOR MACRO-32
ERROR ACTION ROUTINES

J 7

16-SEP-1984 02:04:13 VAX/VMS Macro V04-00
5-SEP-1984 01:48:07 [MACRO.SRC]ERROR.MAR;1

Page 8
(3)

MAC
V04

```
04 0181 322 ERROR_EXIT:
    0181 323          PET
    0182 324
    0182 325          .END
```

MAC\$ERROR
Symbol table

ERROR HANDLER FOR MACRO-32

K 7

16-SEP-1984 07:04:13 VAX/VMS Macro V04-00
5-SEP-1984 01:48:07 [MACRO.SRC]ERROR.MAR;1

Page 9
(3)

MAC
V04

CLI SUBSYS	= 00030000			FLGSM_STOIMF	= 00040000
DSCSA_POINTER	*****	X	03	FLGSM_SYM2COL	= 00000400
DSCSW_LENGTH	*****	X	03	FLGSM_TOCFLG	= 00080000
ERROR_EXIT	00000181	R	03	FLGSM_UPAFLG	= 00000010
ERR_MASK	= 00000FFC			FLGSM_UPDFIL	= 00000080
FABSB_DNS	= 00000035			FLGSM_UPMARG	= 00000040
FABSB_FNS	= 00000034			FLGSM_XCRF	= 80000000
FABSL_DNA	= 00000030			FLGSV_ALLCHR	= 00000000
FABSL_FNA	= 0000002C			FLGSV_BOL	= 00000001
FABSL_NAM	= 00000028			FLGSV_CHKLPND	= 00000014
FABSL_STS	= 00000008			FLGSV_COMPEXPR	= 00000002
FABERR	000000FA	R	03	FLGSV_CONT	= 00000003
FILERR	000000FE	R	03	FLGSV_CRF	= 0000001E
FLGSM_ALLCHR	= 00000001			FLGSV_CRSEEN	= 00000020
FLGSM_BOL	= 00000002			FLGSV_DATRPT	= 00000004
FLGSM_CHKLPND	= 00100000			FLGSV_DBGOUT	= 0000002E
FLGSM_COMPEXPR	= 00000004			FLGSV_CLIMSTR	= 0000002F
FLGSM_CONT	= 00000008			FLGSV_ENDMCH	= 00000005
FLGSM_CRF	= 40000000			FLGSV_EVALEXPR	= 00000006
FLGSM_CRSEEN	= 00000001			FLGSV_EXPOPT	= 00000007
FLGSM_DATRPT	= 00000010			FLGSV_EXTERR	= 00000030
FLGSM_DBGOUT	= 00004000			FLGSV_EXTWRN	= 00000031
FLGSM_DLIMSTR	= 00008000			FLGSV_FIRSTLN	= 00000029
FLGSM_ENDMCH	= 00000020			FLGSV_IFSTAT	= 00000017
FLGSM_EVALEXPR	= 00000040			FLGSV_IIF	= 00000016
FLGSM_EXPOPT	= 00000080			FLGSV_INSERT	= 00000008
FLGSM_EXTERR	= 00010000			FLGSV_IRPC	= 0000001D
FLGSM_EXTWRN	= 00020000			FLGSV_LEXOP	= 00000021
FLGSM_FIRSTLN	= 00000200			FLGSV_LSTXST	= 00000009
FLGSM_IFSTAT	= 00800000			FLGSV_MAC2COL	= 0000002B
FLGSM_IIF	= 00400000			FLGSV_MACL	= 0000000B
FLGSM_INSERT	= 00000100			FLGSV_MACLTB	= 0000001B
FLGSM_IRPC	= 20000000			FLGSV_MACTXT	= 00000010
FLGSM_LEXOP	= 00000002			FLGSV_MEBLST	= 0000000C
FLGSM_LSTXST	= 00000200			FLGSV_MOREARG	= 0000002D
FLGSM_MAC2COL	= 00000800			FLGSV_MOREINP	= 00000023
FLGSM_MACL	= 00000800			FLGSV_NEWPND	= 0000000A
FLGSM_MACLTB	= 08000000			FLGSV_NOREF	= 00000018
FLGSM_MACTXT	= 00010000			FLGSV_NTTYPEPC	= 00000025
FLGSM_MEBLST	= 00001000			FLGSV_NULCHR	= 00000032
FLGSM_MOREARG	= 00002000			FLGSV_OBJXST	= 00000015
FLGSM_MOREINP	= 00000008			FLGSV_OPNDCHK	= 00000028
FLGSM_NEWPND	= 00000400			FLGSV_OPRND	= 0000000D
FLGSM_NOREF	= 01000000			FLGSV_OPTVFLIDX	= 0000002C
FLGSM_NTTYPEPC	= 00000020			FLGSV_ORDLST	= 00000011
FLGSM_NULCHR	= 00040000			FLGSV_P2	= 0000000E
FLGSM_OBJXST	= 00200000			FLGSV_RPTIRP	= 0000001C
FLGSM_OPNDCHK	= 00000100			FLGSV_SEQFIL	= 00000019
FLGSM_OPRND	= 00002000			FLGSV_SKAN	= 0000000F
FLGSM_OPTVFLIDX	= 00001000			FLGSV_SPECOP	= 00000022
FLGSM_ORDLST	= 00020000			FLGSV_SPLALL	= 0000001A
FLGSM_P2	= 00004000			FLGSV_STOIMF	= 00000012
FLGSM_RPTIRP	= 10000000			FLGSV_SYM2COL	= 0000002A
FLGSM_SEQFIL	= 02000000			FLGSV_TOCFLG	= 00000013
FLGSM_SKAN	= 00008000			FLGSV_UPAFLG	= 00000024
FLGSM_SPECOP	= 00000004			FLGSV_UPDFIL	= 00000027
FLGSM_SPLALL	= 04000000			FLGSV_UPMARG	= 00000026

MACSERROR
Symbol table

ERROR HANDLER FOR MACRO-32

L 7

16-SEP-1984 02:04:13 VAX/VMS Macro V04-00
 15-SEP-1984 01:48:07 [MACRO.SRC]ERROR.MAR;1

Page 10
(3)

MAC
V04

```

FLGSV_XCRF          = 0000001F
MACSAB_TPSYM        ***** X 03
MACSERR_CLS_INP     000000EA RG 03
MACSERR_CLS_OUT     000000F2 RG 03
MACSERR_CONNECT     00000081 RG 03
MACSERR_DISCON      00000081 RG 03
MACSERR_FIND        00000081 RG 03
MACSERR_FIND_INP   000000A8 RG 03
MACSERR_GET         00000089 RG 03
MACSERR_INTERN      0000006F RG 03
MACSERR_KEY_WD      00000051 RG 03
MACSERR_LBRÖPEN     00000000 RG 03
MACSERR_MODIFY      00000081 RG 03
MACSERR_NOMEM       0000003E RG 03
MACSERR_NOMEM_0     00000046 RG 03
MACSERR_OPN_INP     000000DA RG 03
MACSERR_OPN_OUT     000000E2 RG 03
MACSERR_PUT         00000091 RG 03
MACSERR_TEXT        0000002D RG 03
MACSGL_FLAGS        ***** X 03
MACSGL_STATUS       ***** X 03
MACSLAST_CHANCE     ***** X 03
MACSMACRO_NAME      00000000 RG 02
MACSPUT_MSG         00000143 RG 03
MLFSK_BKKSIZ       00000177
MLFSK_RSFNLN       = 000000FF
MLFSL_CTINDEX      00000014
MLFSL_MCDEF        00000008
MLFSL_QLINK        00000000
MLFSQ_FNAMDS       0000000C
MLFST_FNAM         00000078
MLFSX_NAMBLK       00000018
NAMSB_ESL          = 0000000B
NAMSB_RSL          = 00000003
NAMSC_BLN          = 00000060
NAMSC_MAXRSS       = 000000FF
NAMSL_ESA          = 0000000C
NAMSL_RSA          = 00000004
OPFSM_LASTOPR     = 00002000
OPFSM_OPTEXP       = 00001000
OPFSV_LASTOPR     = 0000000D
OPFSV_OPTEXP       = 0000000C
RABSL_FAB          = 0000003C
RABSL_STS          = 00000008
RABERR            00000099 R 03
RMSS_EOF           = 0001827A
RMSS_NORMAL        = 00C10001
SHRS_BADLOGICPC    = 00001150
SHRS_BADVALUE      = 00001110
SHRS_CLOSEIN       = 00001050
SHRS_CLOSEOUT      = 00001058
SHRS_OPENIN        = 00001098
SHRS_OPENOUT       = 000010A0
SHRS_READERR       = 000010B0
SHRS_RMSERROR      = 000011C8
SHRS_TEXT          = 00001130
SHRS_WRITEERR      = 000010D0
    
```

```

SRMSL_SAVE_PC     = 00000010
STSSK_ERROR       = 00000002
STSSK_INFO        = 00000003
STSSM_SEVERITY    = 00000007
STSSS_FAC_NO      = 0000000C
STSSV_FAC_NO      = 00000010
SYSSPOTMSG        ***** X 03
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
\$ABSS	00000177 (375.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
MAC\$RO_DATA	0000000D (13.)	02 (2.)	NOPIC USR CON REL GBL NOSHR NOEXE RD NOWRT NOVEC LONG
MAC\$RO_CODE_P4	00000182 (386.)	03 (3.)	NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	36	00:00:00.04	00:00:00.37
Command processing	126	00:00:00.37	00:00:04.17
Pass 1	279	00:00:05.66	00:00:23.88
Symbol table sort	0	00:00:00.81	00:00:04.49
Pass 2	76	00:00:01.13	00:00:04.64
Symbol table output	22	00:00:00.12	00:00:00.12
Psect synopsis output	2	00:00:00.02	00:00:00.61
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	543	00:00:08.16	00:00:38.29

The working set limit was 1500 pages.
49817 bytes (98 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 967 non-local and 7 local symbols.
325 source lines were read in Pass 1, producing 17 object records in Pass 2.
13 pages of virtual memory were used to define 12 macros.

! Macro library statistics !

Macro library name	Macros defined
-\$255\$DUA28:[MACRO.OBJ]MACRO.MLB;1	2
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	11

991 GETS were required to define 11 macros.

There were no errors, warnings or information messages.

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

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

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

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