


```

FFFFFFFFF  AAAAAA  LL      88888888  LL      DDDDDDD  SSSSSSS  TTTTTTTTT  SSSSSSS
FFFFFFFFF  AAAAAA  LL      88888888  LL      DDDDDDD  SSSSSSS  TTTTTTTTT  SSSSSSS
FF         AA      AA  LL      88      88  LL      DD      DD  SS      TT      SS
FF         AA      AA  LL      88      88  LL      DD      DD  SS      TT      SS
FF         AA      AA  LL      88      88  LL      DD      DD  SS      TT      SS
FF         AA      AA  LL      88      88  LL      DD      DD  SS      TT      SS
FFFFFFFFF  AA      AA  LL      88888888  LL      DD      DD  SS      TT      SS
FFFFFFFFF  AA      AA  LL      88888888  LL      DD      DD  SS      TT      SS
FF         AAAAAAAAAA LL      88      88  LL      DD      DD  SS      TT      SS
FF         AAAAAAAAAA LL      88      88  LL      DD      DD  SS      TT      SS
FF         AA      AA  LL      88      88  LL      DD      DD  SS      TT      SS
FF         AA      AA  LL      88      88  LL      DD      DD  SS      TT      SS
FF         AA      AA  LLLLLLLLLL 88888888 LI.LLLLLLLL DDDDDDD  SSSSSSS  TT      SSSSSSS  ....
FF         AA      AA  LLLLLLLLLL 88888888 LLLLLLLLLL DDDDDDD  SSSSSSS  TT      SSSSSSS  ....

```

```

LL         IIIIII  SSSSSSS
LL         IIIIII  SSSSSSS
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  SSSSSSS
LLLLLLLLLL IIIIII  SSSSSSS

```

FALBLDSTS
Table of contents

- BUILD DAP STATUS MESSAGE

K 13

16-SEP-1984 01:38:54 VAX/VMS Macro V04-00

Page 0

FAL
V04

(2) 48
(3) 75

DECLARATIONS
FALSENCODE_STS - BUILD DAP STATUS MESSAGE

```

0000 1      .TITLE FALBLDSTS - BUILD DAP STATUS MESSAGE
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: FAL (DECnet File Access Listener)
0000 31 :
0000 32 : Abstract: This module builds the DAP Status message.
0000 33 :
0000 34 : Environment: VAX/VMS, user mode
0000 35 :
0000 36 : Author: James A. Krycka,      Creation Date: 16-JUN-1977
0000 37 :
0000 38 : Modified By:
0000 39 :
0000 40 :      V03-002 JAK0137      J A Krycka      12-MAR-1984
0000 41 :      Use $CHECK_STATUS instead of $CHECK_SS and other minor cleanup.
0000 42 :
0000 43 :      V03-001 JAK0104      J A Krycka      15-MAR-1983
0000 44 :      !f logging in enabled, omit display of DAP status code that is
0000 45 :      equivalent to RMSS_EOF.
0000 46 :--

```

```
0000 48      .SBTTL  DECLARATIONS
0000 49
0000 50 :
0000 51 : Include Files:
0000 52 :
0000 53
0000 54      $DAPHDRDEF      ; Define DAP message header
0000 55      $DAPCNFDEF     ; Define DAP Configuration message
0000 56      $DAPSTSDEF     ; Define DAP Status message
0000 57      $FALWRKDEF    ; Define FAL Work Area symbols
0000 58      $RABDEF       ; Define Record Access Block sym**
0000 59
0000 60 :
0000 61 : Macros:
0000 62 :
0000 63 :     None
0000 64 :
0000 65 : Equated Symbols:
0000 66 :
0000 67
0000 68      ASSUME  FAL$Q_FLG EQ 0
0000 69
0000 70 :
0000 71 : Own Storage:
0000 72 :
0000 73 :     None
```

```
0000 0000 75 .SBTTL FALSENCODE_STS - BUILD DAP STATUS MESSAGE
0000 0000 76 .PSECT FAL$CODE NOSHR,EXE,RD,NOWRT,BYTE
0000 0000 77
0000 0000 78 :++
0000 0000 79 : Functional Description:
0000 0000 80 :
0000 0000 81 : FALSENCODE_STS builds a DAP Status message based on the class of DAP
0000 0000 82 : success or failure condition.
0000 0000 83 :
0000 0000 84 : Calling Sequence:
0000 0000 85 :
0000 0000 86 : BSBW FALSENCODE_STS
0000 0000 87 :
0000 0000 88 : Input Parameters:
0000 0000 89 :
0000 0000 90 : R2 DAP MACCODE value (in bits 0-3)
0000 0000 91 : R8 Address of FAL work area
0000 0000 92 : R9 Address of DAP control block
0000 0000 93 : R10 Address of File Access Block
0000 0000 94 : R11 Address of RAB control block
0000 0000 95 :
0000 0000 96 : and for file access status codes:
0000 0000 97 :
0000 0000 98 : R0 RMS-32 completion code
0000 0000 99 : R1 Secondary status value (STV of FAB or RAB)
0000 0000 100 :
0000 0000 101 : and for DAP protocol errors:
0000 0000 102 :
0000 0000 103 : R0 DAP field ID code (in bits 0-5)
0000 0000 104 : R1 DAP message type (in bits 0-5)
0000 0000 105 :
0000 0000 106 : and for message synchronization errors:
0000 0000 107 :
0000 0000 108 : R0 Not used
0000 0000 109 : R1 DAP message type (in bits 0-5)
0000 0000 110 :
0000 0000 111 : Implicit Inputs:
0000 0000 112 :
0000 0000 113 : FAL$V_LAST_MSG
0000 0000 114 : FAL$V_RET_RFA
0000 0000 115 : FAL$V_RET_REC�N
0000 0000 116 : FAL$V_RET_STV
0000 0000 117 : RAB$L_BKT
0000 0000 118 : RAB$W_RFA
0000 0000 119 :
0000 0000 120 : Output Parameters:
0000 0000 121 :
0000 0000 122 : R0-R5 Destroyed
0000 0000 123 :
0000 0000 124 : Implicit Outputs:
0000 0000 125 :
0000 0000 126 : FAL$V_LAST_MSG cleared
0000 0000 127 : FAL$V_RET_RFA cleared
0000 0000 128 : FAL$V_RET_REC�N cleared
0000 0000 129 : FAL$V_RET_STV cleared
0000 0000 130 :
0000 0000 131 : Completion Codes:
```

```

0000 132 :
0000 133 :      None
0000 134 :
0000 135 :      Side Effects:
0000 136 :
0000 137 :      None
0000 138 :
0000 139 :      --
0000 140 :
0000 141 FALSENCODE_STS::      : Control point
0000 142 :
0000 143      ASSUME DAPS_PENDING EQ 0
0000 144      ASSUME DAPS_SUCCESS EQ 1
0000 145      ASSUME DAPS_UNSUPPORT EQ 2
0000 146      ASSUME DAPS_FILE_OPEN EQ 4
0000 147      ASSUME DAPS_FILE_XFER EQ 5
0000 148      ASSUME DAPS_WARNING EQ 6
0000 149      ASSUME DAPS_FILE_CLOS EQ 7
0000 150      ASSUME DAPS_FORMAT EQ 8
0000 151      ASSUME DAPS_INVALID EQ 9
0000 152      ASSUME DAPS_MSG_SYNC EQ 10
0000 153 :
0000 154 :
0000 155 :      Examine DAP MACCODE parameter to determine which class of message it is.
0000 156 :
0000 157 :
0000 158      $CASEB SELECTOR=R2,-      : MACCODE:
0000 159      DISPI=<-      :
0000 160      FILE_ACCESS-      : File operation pending
0000 161      FILE_ACCESS-      : File operation successful
0000 162      PROTOCOL-      : Request not supported
0000 163      BUG-      : Undefined value
0000 164      FILE_ACCESS-      : Errors related to opening a file
0000 165      FILE_ACCESS-      : Errors related to file transfer
0000 166      BUG-      : Undefined value
0000 167      FILE_ACCESS-      : Errors related to closing a file
0000 168      PROTOCOL-      : Incorrect message format
0000 169      PROTOCOL-      : Invalid field value
0000 170      MSG_OUT_OF_SEQ-      : Unexpected message for state
0000 171      >      :
0000 172 BUG:      : Fall thru ...
0000 173 :
0000 174 :
0000 175 :      Process file access error. perform RMS-to-DAP status code look-up to obtain
0000 176 :      DAP MICCODE value.
0000 177 :
0000 178 :      Note: If the RMS code does not match, a DAP MICCODE of zero is returned
0000 179 :      which means unknown type of error.
0000 180 :
0000 181 :
0000 182 FILE_ACCESS:      :
51 55 51 DO 001A 183      MOVL R1,R5      : Save STV value in R5
0000 81 CF DE 001D 184      MOVAL W^FAL$RMS_TO_DAP,R1      : Get conversion table address
81 50 B1 0022 185 10$: CMPW R0,(R1)+      : Does RMS code match?
04 13 0025 186      BEQL 20$      : Yes, finish up
81 B5 0027 187      TSTW (R1)+      : Is this end of table?
F7 12 0029 188      BNEQ 10$      : No, try again

```

```

50 61 3C 002B 189 20$:  MOVZWL (R1),R0          ; Store equivalent DAP status code
    0A 11 002E 190          BRB      ADD_MACCODE      ; Continue
    0030 191
    0030 192 ;
    0030 193 ; Form MICCODE value for DAP protocol error.
    0030 194 ;
    0030 195 ;
    50 06 06 51  F0 0030 196 PROTOCOL:          ;
    03 11 0030 197          INSV   R1,#6,#6,R0      ; Add message type to field id
    0035 198          BRB      ADD_MACCODE      ; Continue
    0037 199
    0037 200 ;
    0037 201 ; Get MICCODE value for message-out-of-sequence error.
    0037 202 ;
    0037 203 ;
    50 51  D0 0037 204 MSG_OUT_OF_SEQ:          ;
    0037 205          -MOVL   R1,R0              ; MACCODE = message type value
    003A 206
    003A 207 ;
    003A 208 ; Combine MACCODE value with MICCODE value to form DAP status code.
    003A 209 ;
    003A 210 ;
    50 04 0C 52  F0 003A 211 ADD_MACCODE:          ;
    54 50  D0 003F 212          INSV   R2,#12,#4,R0  ; Assemble DAP status code in bits 00-15
    0042 213          MOVL   R0,R4              ; and save value in R4
    0042 214
    0042 215 ;
    0042 216 ; Begin constructing the Status message.
    0042 217 ;
    0042 218 ;
    50 09  D0 0042 219 SEND_STS:                ;
    FF 30 0045 220          MOVL   #DAP$K_STS_MSG,R0 ; Get message type value
    83 54  B0 0048 221          BSBW   FALS$BUILD_READ ; Construct message header
    004B 222          MOVW   R4,(R3)+          ; Store status code field
    004B 223
    004B 224 ;
    004B 225 ; The required fields have been assembled (header plus STSCODE).
    004B 226 ; Now assemble the optional fields (RFA, RECNUM, and STV) as requested.
    004B 227 ;
    004B 228 ; Note: If FALS$V_RET_RECNUM is set, then get value from RABS$L_BKT.
    004B 229 ;       If FALS$V_RET_RFA is set, then get value from RABS$L_RFA0 and RABS$W_RFA4.
    004B 230 ;       If FALS$V_RET_STV is set, then get value from R5.
    004B 231 ;       Finally, clear all three bits.
    004B 232 ;
    004B 233 ;
    004B 234          ASSUME  FALS$V_RET_RECNUM EQ FALS$V_RET_RFA+1
    004B 235          ASSUME  FALS$V_RET_STV EQ FALS$V_RET_RECNUM+1
    004B 236
    04 28 0A  E0 004B 237          BBS     #DAP$V_RANRFA,-          ; Branch if partner supports
    004D 238          DAP$Q_SYSCAP(R9),10$          ; access by RFA
    50 68 03 19  EF 0050 239          $CLRBIT #FALS$V_RET_RFA,(R8) ; Disable request to return RFA
    30 13 0054 240 10$:  EXTZV  #FALS$V_RET_RFA,#3,(R8),R0 ;
    005B 241          BEQL   40$                  ; Branch if RFA, RECNUM, and STV
    005B 242          ;                          ; are not wanted
    005B 243          CLRB   (R3)+                ; Assume RFA is not wanted
    0C 68 19  E5 005D 244          BBCC  #FALS$V_RET_RFA,(R8),20$ ; Branch if RFA is not wanted
    FF A3 06 90 0061 245          MOVB   #6,-1(R3)          ; Store RFA as an image field

```



```

50 83 10 AB DO 0065 246      MOVL  RAB$L_RFA0(R11),(R3)+  ; Copy 6-byte record file address
    83 14 AB B0 0069 247      MOVW  RAB$W_RFA4(R11),(R3)+  ; value to DAP message
    68 02 1A EF 006D 248 20$: EXTZV #FALS$V_RET_RECNUM,#2,(R8),R0
    17 13 0072 249      BEQL  40$                    ; Branch if RECNUM and STV
    0074 250      ; are not wanted
    51 D4 0074 251      CLRL  R1                    ; Assume RECNUM is not wanted
    04 68 1A E5 0076 252      BBCC  #FALS$V_RET_RECNUM,(R8),30$ ; Branch if RECNUM is not wanted
    51 38 AB DO 007A 253      MOVL  RAB$L_BKT(R11),R1    ; Get bucket value to use for RECNUM
    FF7F' 30 007E 254 30$: BSBW  FALS$CVT_BN4_IMG ; Store RECNUM as an image field
    06 68 1B E5 0081 255      BBCC  #FALS$V_RET_STV,(R8),40$ ; Branch if STV is not wanted
    51 55 DO 0085 256      MOVL  R5,R1                    ; Get status value
    FF75' 30 0088 257      BSBW  FALS$CVT_BN4_IMG ; Store STV as an image field
    FF72' 30 008B 258 40$: BSBW  FALS$BUIED_TAIL ; Finish building message
    008E 259      ;
    008E 260      ; Output status code to print file.
    008E 261      ;
    008E 262      ;
    29 68 20 E1 008E 264      BBC   #FALS$V_LOG_NAM,(R8),50$ ; Branch if logging disabled
    1095 8F 54 B1 0092 265      CMPW  R4,#^XT095-          ; Omit display if DAP status code is
    22 13 0097 266      BEQL  50$                    ; equivalent to RMSS_SUC
    5027 8F 54 B1 0099 267      CMPW  R4,#^X5027         ; Omit display if DAP status code is
    1B 13 009E 268      BEQL  50$                    ; equivalent to RMSS_EOF
    00A0 269      $FAO_S  CTRSTR=W^FALS$GQ_STATUS- ; Format the error message
    00A0 270      ; OUTLEN=W^FALS$GW_PRTLEN1-
    00A0 271      ; OUTBUF=W^FALS$GQ_PRTBUF1-
    00A0 272      ; P1=R4
    FF45' 30 00B5 273      $CHECK_STATUS ; DAP status word
    00B8 274      BSBW  FALS$PRINT_FAO ; Check status code
    05 00BB 275 50$: RSB   ; Print message
    00BC 276      ; Exit
    00BC 277      .END ; End of module

```

\$\$COUNT	= 0000000B			FAL\$ENCODE STS	00000000	RG	02
\$\$T2	= 00000004			FAL\$GQ_PRTBUF1	*****	X	02
ADD_MACCODE	0000003A	R	02	FAL\$GQ_STATUS	*****	X	02
BUG	0000001A	R	02	FAL\$GW_PRTLEN1	*****	X	02
DAP\$B_BITCNT	00000035			FAL\$K_WRKBLN	00002000		
DAP\$B_DECVER	00000047			FAL\$L_ALLXAB	00000C00		
DAP\$B_ECONUM	00000045			FAL\$L_ALLXABINI	00000074		
DAP\$B_FILESYS	00000043			FAL\$L_CHAIN NXT	0000007C		
DAP\$B_FLAGS	00000031			FAL\$L_DATXAB	00000320		
DAP\$B_LEN256	00000034			FAL\$L_FAB	00000200		
DAP\$B_LENGTH	00000033			FAL\$L_FAB2	00000800		
DAP\$B_OSTYPE	00000042			FAL\$L_FHCXAB	000002F4		
DAP\$B_STREAMID	00000032			FAL\$L_FOP	000001F8		
DAP\$B_TYPE	00000030			FAL\$L_KEYNAM	00001C00		
DAP\$B_USRNUM	00000046			FAL\$L_KEYXAB	00001000		
DAP\$B_USRVER	00000048			FAL\$L_KEYXABINI	00000078		
DAP\$B_VERNUM	00000044			FAL\$L_NAM	00000294		
DAP\$K_STS MSG	= 00000009			FAL\$L_NAM2	00000850		
DAP\$L_RECNUM2	00000048			FAL\$L_NUMBER	000001FC		
DAP\$L_STV	0000004C			FAL\$L_PROXAB	0000034C		
DAP\$M_BITCNT	= 00000008			FAL\$L_RAB	00000250		
DAP\$M_SEGMENT	= 00000040			FAL\$L_RCVBUF	0000005C		
DAP\$M_TMP1\$	= 00000010			FAL\$L_RDTXAB	000003B0		
DAP\$M_TMP2\$	= 00000080			FAL\$L_RMS_PTR	0000006C		
DAP\$Q_STX	00000050			FAL\$L_STB	000000C0		
DAP\$Q_SYSCAP	00000028			FAL\$L_SUMXAB	000003A4		
DAP\$Q_SYSPEC	00000038			FAL\$L_TEMP	000003F4		
DAP\$V_RANRFA	= 0000000A			FAL\$L_USE_SC1	000000A8		
DAP\$W_BUFSIZ	00000040			FAL\$L_USE_SC2	000000AC		
DAP\$W_RFA	00000042			FAL\$L_USE_VER	000000A4		
DAP\$W_STSCODE	00000040			FAL\$PRINT_FAO	*****	X	02
DAP\$_FILE_CLOS	= 00000007			FAL\$Q_BLD	00000050		
DAP\$_FILE_OPEN	= 00000004			FAL\$Q_DIRNAME	00000088		
DAP\$_FILE_XFER	= 00000005			FAL\$Q_FALLG	00000090		
DAP\$_FORMAT	= 00000008			FAL\$Q_FLG	00000000		
DAP\$_INVALID	= 00000009			FAL\$Q_MBX	00000038		
DAP\$_MSG_SYNC	= 0000000A			FAL\$Q_MBXIOSB	00000030		
DAP\$_PENDING	= 00000000			FAL\$Q_RCV	00000040		
DAP\$_SUCCESS	= 00000001			FAL\$Q_RCVIOSB	00000020		
DAP\$_UNSUPPORT	= 00000002			FAL\$Q_RMS	00000064		
DAP\$_WARNING	= 00000006			FAL\$Q_STATE CTX	00000008		
FAL\$BUILD_HEAD	*****	X	02	FAL\$Q_SYSNET	00000098		
FAL\$BUILD_TAIL	*****	X	02	FAL\$Q_TEMP	000003F8		
FAL\$B_ACCFUNC	000001F6			FAL\$Q_VOLNAME	00000080		
FAL\$B_ACCOPT	000001F5			FAL\$Q_XMT	00000048		
FAL\$B_DATATYPE	000001F4			FAL\$Q_XMTIOSB	00000028		
FAL\$B_DISABLE	00000006			FAL\$RMS TO_DAP	*****	X	02
FAL\$B_ENABLE	00000005			FAL\$T_DAP	00000100		
FAL\$B_LOGGING	00000004			FAL\$T_DIRNAME	00001F00		
FAL\$B_MISCOPT	00000007			FAL\$T_EXPAND	00000500		
FAL\$B_RAC	000001F7			FAL\$T_EXPAND2	00000A00		
FAL\$B_RBK CACHE	00000012			FAL\$T_FALLG	00001C00		
FAL\$B_RCVBUF IDX	00000011			FAL\$T_FILESPEC	00000400		
FAL\$B_VALUE	00000010			FAL\$T_FILESPEC2	00000900		
FAL\$CHECK STATUS	*****	X	02	FAL\$T_KEYBUF	00000700		
FAL\$CVT BR4 IMG	*****	X	02	FAL\$T_MBXBUF	00001980		
FAL\$C_WRKBLN	00002000			FAL\$T_PRTBUF1	00001A00		

FALBLDSTS
Symbol table

- BUILD DAP STATUS MESSAGE

F 14

16-SEP-1984 01:38:54 VAX/VMS Macro V04-00
5-SEP-1984 01:16:31 [FAL.SRC]FALBLDSTS.MAR;1

Page 8
(3)

FAL
V04

```

FALST_PRTBUF2      00001B00
FALST_RESULT      00000600
FALST_RESULT2     00000800
FALST_SYSNET      00001D00
FALST_VOLNAME     00001E00
FALSV_LOG_NAM     = 00000020
FALSV_RET_REC�    = 0000001A
FALSV_RET_RFA     = 00000019
FALSV_RET_STV     = 0000001B
FALSW_DAPBUFSIZ   0000001A
FALSW_DISPLAY     00000070
FALSW_LNKCHN      0000001C
FALSW_MBXCHN      0000001E
FALSW_QIOBUFSIZ   00000018
FALSW_RECEIVED    00000072
FALSW_USE_DBS     000000A0
FALSW_USE_SYS     000000A2
FILE_ACCESS       0000001A R      02
MSG_OUT_OF_SEQ    00000037 R      02
PROTOCOL          00000030 R      02
RABSL_BKT         = 00000038
RABSL_RFA0        = 00000010
RABSW_RFA4        = 00000014
SEND_STS          00000042 R      02
SYSSFAO          ***** X      02
  
```

! 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	00002000 (8192.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
FAL\$CODE	000000BC (188.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.04	00:00:01.13
Command processing	105	00:00:00.38	00:00:01.85
Pass 1	225	00:00:04.29	00:00:19.27
Symbol table sort	0	00:00:00.49	00:00:00.95
Pass 2	63	00:00:00.84	00:00:02.40
Symbol table output	17	00:00:00.07	00:00:00.25
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	443	00:00:06.12	00:00:25.86

The working set limit was 1050 pages.
31830 bytes (63 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 532 non-local and 9 local symbols.
277 source lines were read in Pass 1, producing 13 object records in Pass 2.
20 pages of virtual memory were used to define 18 macros.

! Macro library statistics !

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[FAL.OBJ]FAL.MLB;1	8
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	15

698 GETS were required to define 15 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:FALBLDSTS/OBJ=OBJ\$:FALBLDSTS MSRC\$:FALBLDSTS/UPDATE=(ENHS:FALBLDSTS)+LIB\$:FAL/LIB

Grid of 100 terminal window screenshots (10 rows by 10 columns). Each window displays a different VAX/VMS command and its output. The windows are arranged in a grid, with some larger windows in the top-left and bottom-right corners. The text in the windows is small and difficult to read, but some titles are visible:

- Top-left: FALACTION LIS
- Top-right: FALDAPRC LIS
- Middle-left: FALMACROS MAR
- Middle-right: FALBLDXAB LIS
- Bottom-left: FALDEF MDL
- Bottom-right: FALDAP10 LIS
- Bottom-center: FALACTMSG LIS

The screenshots show various system messages, file listings, and command outputs, demonstrating the VAX/VMS operating system's capabilities.