


```

FFFFFFFFF      AAAAAA      LL      88888888      LL      DDDDDDD      AAAAAA      TTTTTTTTTT      TTTTTTTTTT
FFFFFFFFF      AAAAAA      LL      88888888      LL      DDDDDDD      AAAAAA      TTTTTTTTTT      TTTTTTTTTT
FF            AA      AA      LL      88      88      LL      DD      DD      AA      AA      TT      TT
FF            AA      AA      LL      88      88      LL      DD      DD      AA      AA      TT      TT
FF            AA      AA      LL      88      88      LL      DD      DD      AA      AA      TT      TT
FF            AA      AA      LL      88      88      LL      DD      DD      AA      AA      TT      TT
FFFFFFFFF      AA      AA      LL      88888888      LL      DD      DD      AA      AA      TT      TT
FFFFFFFFF      AA      AA      LL      88888888      LL      DD      DD      AA      AA      TT      TT
FF            AAAAAAAAAA      LL      88      88      LL      DD      DD      AAAAAAAAAA      TT      TT
FF            AAAAAAAAAA      LL      88      88      LL      DD      DD      AAAAAAAAAA      TT      TT
FF            AA      AA      LL      88      88      LL      DD      DD      AA      AA      TT      TT
FF            AA      AA      LL      88      88      LL      DD      DD      AA      AA      TT      TT
FF            AA      AA      LLLLLLLLLL      88888888      LLLLLLLLLL      DDDDDDD      AA      AA      TT      TT
FF            AA      AA      LLLLLLLLLL      88888888      LLLLLLLLLL      DDDDDDD      AA      AA      TT      TT

```



```

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
LLLLLLLLLLL      IIIIII      SSSSSSS
LLLLLLLLLLL      IIIIII      SSSSSSS

```

FAL
VAX

Pha

Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
649
The
377
22

....
....
....
....

Mac

-S2
-S2
TOT

135

The
MAC

FALBLDATT
Table of contents

- BUILD DAP ATTRIBUTES MESSAGE

J 12

16-SEP-1984 01:38:03 VAX/VMS Macro V04-00

Page 0

**F

(2) 55
(3) 88
(4) 304

DECLARATIONS
FALSENCODE_ATT
FALSENCODE_NAM

```

0000 1 .TITLE FALBLDATT - BUILD DAP ATTRIBUTES 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 Attributes and Name messages.
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 V03-004 JAK0144 J A Krycka 11-APR-1984
0000 41 Minor cleanup.
0000 42
0000 43 V03-003 KRM0096 K Malik 06-Apr-1983
0000 44 Modified FALSENCODE_NAM to support rename operation
0000 45 (to use NAM2 for 2nd resultant Name message).
0000 46
0000 47 V03-002 KRM0084 K Malik 23-Mar-1983
0000 48 Add support for STMLF and STMCR formats.
0000 49
0000 50 V03-001 KRM0068 K Malik 23-Nov-1982
0000 51 Modified FALSENCODE_NAM to support rename operation.
0000 52
0000 53 --

```

```
0000 55          .SBTTL  DECLARATIONS
0000 56
0000 57          :
0000 58          : Include Files:
0000 59          :
0000 60
0000 61          $DAPPLGDEF          ; Define DAP prologue symbols
0000 62          $DAPHDRDEF         ; Define DAP message header
0000 63          $DAPATTDEF         ; Define DAP Attributes message
0000 64          $DAPACCDEF         ; Define DAP Access message
0000 65          $DAPNAMDEF        ; Define DAP Name message
0000 66          $DEVDEF           ; Define device characteristics symbols
0000 67          $FABDEF           ; Define File Access Block symbols
0000 68          $FALWRKDEF        ; Define FAL Work Area symbols
0000 69          $NAMDEF           ; Define Name Block symbols
0000 70          $XABDEF           ; Define symbols common to all XABs
0000 71          $XABFHCDEF        ; Define File Header Char XAB symbols
0000 72
0000 73          :
0000 74          : Macros:
0000 75          :
0000 76          :     None
0000 77          :
0000 78          : Equated Symbols:
0000 79          :
0000 80
0000 81          ASSUME  DAP$Q_DCODE_FLG EQ 0
0000 82          ASSUME  FAL$Q_FLG EQ 0
0000 83
0000 84          :
0000 85          : Own Storage:
0000 86          :
```

```

0000 88      .SBTTL  FALSENCODE_ATT
0000 89      .PSECT  FALS$CODE      NOSHR,EXE,RD,NOWRT,BYTE
0000 90
0000 91      :++
0000 92      : Functional Description:
0000 93      :
0000 94      :     FALSENCODE_ATT builds the DAP Attributes message.
0000 95      :
0000 96      : Calling Sequence:
0000 97      :
0000 98      :     BSBW  FALSENCODE_ATT
0000 99      :
0000 100     : Input Parameters:
0000 101     :
0000 102     :     R8      Address of FAL work area
0000 103     :     R9      Address of DAP control block
0000 104     :     R10     Address of FAB
0000 105     :     R11     Address of RAB
0000 106     :
0000 107     : Implicit Inputs:
0000 108     :
0000 109     :     DAP$V_GEQ_V54
0000 110     :     DAP$V_GEQ_V56
0000 111     :     DAP$V_GEQ_V70
0000 112     :     DAP$V_VAXVMS
0000 113     :     FAB fields
0000 114     :     FALS$B_ACCFUNC
0000 115     :     FALS$B_DATATYPE
0000 116     :
0000 117     : Output Parameters:
0000 118     :
0000 119     :     R0-R7  Destroyed
0000 120     :
0000 121     : Implicit Outputs:
0000 122     :
0000 123     :     None
0000 124     :
0000 125     : Completion Codes:
0000 126     :
0000 127     :     None
0000 128     :
0000 129     : Side Effects:
0000 130     :
0000 131     :     None
0000 132     :
0000 133     :--
0000 134     :
50   02   D0 0000 135 FALSENCODE_ATT::      : Control point
   FFFA' 30 0000 136      MOVL  #DAP$K_ATT_MSG,R0  : Get message type value
0003 137      BSBW  FALS$BUILD_READ      : Construct message header
0006 138
0006 139 :+
0006 140 : Determine which fields to return as follows:
0006 141 : (1) always send the ORG, RFM, RAT, MRS, ALQ, DEQ, and FOP fields.
0006 142 : (2) send the DATATYPE and BLS fields only if they do not equal their
0006 143 :     default values as defined in the DAP specification.
0006 144 : (3) send the BKS field only if ORG = REL or IDX; and MRN only if ORG = REL.

```

```

0006 145 : (4) send the FSZ field only if RFM = VFC.
0006 146 : (5) send the DEV field only if accessing node is VAX/VMS or it is
0006 147 : implemented to DAP V5.6 or later.
0006 148 : (6) send the LRL, HBK, EBK, FFB, SDN fields if this is not a create
0006 149 : function (i.e., its an open or directory function) and the accessing
0006 150 : node supports DAP V5.4 or later.
0006 151 : (7) never send the RUNSYS field.
0006 152 :-
0006 153 :-
51 0000186E 8F D0 0006 154      MOVL    #<<DAP$M_ORG>!--      : Always send ORG, RFM, RAT, MRS, ALQ
0000 155      <DAP$M_RFM>!--      : DEQ, and FOP fields
0000 156      <DAP$M_RAT>!--
0000 157      <DAP$M_MRS>!--
0000 158      <DAP$M_ALQ1>!--
0000 159      <DAP$M_DEQ1>!--
0000 160      <DAP$M_FOP1>!--
0000 161      0>,R1
      01F6 C8 91 0000 162      CMPB    FALS$B_ACCFUNC(R8),-      : Default DATATYPE field on
0006 163      #DAP$R_DIR_LIST      : directory_list function as
0006 164      10$      : FALS$B_ACCFUNC = 0
      01F4 C8 91 0014 165      CMPB    FALS$B_DATATYPE(R8),-      : Branch if DATATYPE = default value
0006 166      #DAP$R_DATATYP_D
0006 167      10$
0006 168      $SETBIT #DAP$V_DATATYPE,R1      : Send DATATYPE
      3C AA B1 001F 169 10$:      CMPW    FABS$W_BLS(R10),-      : Branch if BLS = default value
0200 8F 04 13 0022 170      #DAP$R_BLS_D
0006 171      20$
0006 172      $SETBIT #DAP$V_BLS,R1      : Send BLS
00 1D AA 91 002B 173 20$:      CMPB    FABS$B_ORG(R10),#FABS$C_SEQ
0006 174      30$
0006 175      $SETBIT #DAP$V_BKS,R1      : Send BKS if org = REL or IDX
10 1D AA 91 0035 176 30$:      CMPB    FABS$B_ORG(R10),#FABS$C_REL
0006 177      40$
0006 178      $SETBIT #DAP$V_MRN,R1      : Send MRN if ORG = REL
03 1F AA 91 003F 179 40$:      CMPB    FABS$B_RFM(R10),#FABS$C_VFC
0006 180      50$
0006 181      $SETBIT #DAP$V_FSZ,R1      : Send FSZ if RFM = VFC
04 69 34 E0 0049 182 50$:      BBS     #DAP$V_VAXVMS,(R9),60$      : Branch if partner is VAX/VMS
04 69 24 E1 004D 183 70$:      BBC     #DAP$V_GEQ_V56,(R9),70$      : Branch if partner uses DAP before V5.6
0006 184 60$:      $SETBIT #DAP$V_DEV,R1      : Send DEV
0006 185 70$:      CMPB    FALS$B_ACCFUNC(R8),-      : Branch if create function
0006 186      #DAP$R_CREATE
0006 187      80$
0006 188      or
51 07 69 23 E1 005C 188 80$:      BBC     #DAP$V_GEQ_V54,(R9),80$      : Branch if partner uses DAP before V5.4
001F0000 8F C8 0060 189      BISL2  #<<DAP$M_LRL>!--      : Send LRL, HBK, EBK, FFB, and SBN
0006 190      <DAP$M_HBK>!--      : fields
0006 191      <DAP$M_EBK>!--
0006 192      <DAP$M_FFB>!--
0006 193      <DAP$M_SBN>!--
0006 194      0>,R1
      56 51 D0 0067 195 80$:      MOVL    R1,R6      : Save the send field flags
      FF93' 30 006A 196      BSBW   FALS$CVT_BN4_EXT      : Store ATTMENU as an extensible field
0006 197
0006 198 :+
0006 199 : Now store the designated fields in the order specified by ATTMENU.
0006 200 :-
0006 201

```

```

05 56 00 E1 006D 202 BSC #DAPSV_DATATYPE,R6,100$ ;
83 01F4 C8 90 0071 203 MOVW FALS$B_DATATYPE(R8),(R3)+ ; Store DATATYPE field
83 1D AA 90 0076 204 100$: MOVW FABS$B_ORG(R10),(R3)+ ; Store ORG field
83 1F AA 90 007A 205 MOVW FABS$B_RFM(R10),(R3)+ ; Store RFM field
0A 69 26 E0 007E 206 BBS #DAPSV_GEQ_V70,(R9),110$ ; Branch if partner uses DAP V7.0
; or greater
04 1F AA 91 0082 207 CMPB FABS$B_RFM(R10),#FABS$C_STM ; Is this format valid for DAP
; 110$ ; version 6.0 or earlier?
FF A3 00 90 0088 208 BLEQU #FABS$C_UDF,-1(R3) ; If not, say format is undefined
83 1E AA 90 008C 211 110$: MOVW FABS$B_RAT(R10),(R3)+ ; Store RAT field
0A 69 34 E0 0090 212 BBS #DAPSV_VAXVMS,(R9),120$ ; Branch if partner is VAX/VMS
04 1F AA 91 0094 213 CMPB FABS$B_RFM(R10),#FABS$C_STM ; Branch if not stream format
; 120$
FF A3 10 90 009A 215 MOVW #DAPSM_EMBEDDED,-1(R3) ; If it is, say cc is embedded
04 56 04 E1 009E 216 120$: BBC #DAPSV_BLS,R6,130$ ;
83 3C AA B0 00A2 217 MOVW FABS$W_BLS(R10),(R3)+ ; Store BLS field
83 36 AA B0 00A6 218 130$: MOVW FABS$W_MRS(R10),(R3)+ ; Store MRS field
51 10 AA D0 00AA 219 MOVL FABS$L_ALQ(R10),R1 ; Get ALQ value
; FF4F' 30 00AE 220 BSBW FALS$CVT_BN4_IMG ; Store ALQ as an image field
04 56 07 E1 00B1 221 BBC #DAPSV_BKS,R6,140$ ;
83 3E AA 90 00B5 222 MOVW FABS$B_BKS(R10),(R3)+ ; Store BKS field
04 56 08 E1 00B9 223 140$: BBC #DAPSV_FSZ,R6,150$ ;
83 3F AA 90 00BD 224 MOVW FABS$B_FSZ(R10),(R3)+ ; Store FSZ field
07 56 09 E1 00C1 225 150$: BBC #DAPSV_MRN,R6,160$ ;
51 38 AA D0 00C5 226 MOVL FABS$L_MRN(R10),R1 ; Get MRN value
; FF34' 30 00C9 227 BSBW FALS$CVT_BN4_IMG ; Store MRN as an image field
83 14 AA B0 00CC 228 160$: MOVW FABS$W_DEQ(R10),(R3)+ ; Store DEQ field
; 00D0 229
; 00D0 230 ;+
; 00D0 231 ; In constructing the DAP FOP field, take advantage of the fact that
; 00D0 232 ; RMS-32 may modify only the CTG, CBT, RCK, and WCK bits on $OPEN (none
; 00D0 233 ; are modified on $CREATE).
; 00D0 234 ;:-
; 00D0 235
52 52 01F8 C8 D0 00D0 236 MOVL FALS$L_FOP(R8),R2 ; Get partner supplied FOP bits
52 0080C080 8F CA 00D5 237 BICL2 #<<DAPSM_CTG>!-- ; Clear bits that may have changed
; 00DC 238 <DAPSM_CBT>!--
; 00DC 239 <DAPSM_RCK>!--
; 00DC 240 <DAPSM_WCK>!--
; 00DC 241 0>,R2
51 04 AA D0 00DC 242 MOVL FABS$L_FOP(R10),R1 ; Get FOP bits returned by RMS
; 00E0 243 $MAPBIT FABS$V_CTG,DAPSV_CTG ; Map CTG bit
; 00E8 244 $MAPBIT FABS$V_RCK,DAPSV_RCK ; Map RCK bit
; 00F0 245 $MAPBIT FABS$V_WCK,DAPSV_WCK ; Map WCK bit
08 69 34 E1 00F8 246 BBC #DAPSV_VAXVMS,(R9),170$ ; Branch if partner is not VAX/VMS
; 00FC 247 $MAPBIT FABS$V_CBT,DAPSV_CBT ; Map CBT bit
51 52 D0 0104 248 170$: MOVL R2,R1 ; Move data to correct register
; FEF6' 30 0107 249 BSBW FALS$CVT_BN4_EXT ; Store FOP as an extensible field
; 010A 250
; 010A 251 ;+
; 010A 252 ; Map and store the DEV field.
; 010A 253 ;:-
; 010A 254
03 56 0E E0 010A 255 BBS #DAPSV_DEV,R6,180$ ; Determine if this field should be
; 00D4 31 010E 256 BRW 190$ ; included in message
51 40 AA D0 0111 257 180$: MOVL FABS$L_DEV(R10),R1 ; Get DEV bits returned by RMS
; 52 D4 0115 258 CLRL R2 ; Clear corresponding DAP bits

```

```

0117 259 $MAPBIT DEV$V_REC,DAP$V_DEVREC : Map REC bit
011F 260 $MAPBIT DEV$V_CCL,DAP$V_DEVCCCL : Map CCL bit
0127 261 $MAPBIT DEV$V_TRM,DAP$V_DEVTRM : Map TRM bit
012F 262 $MAPBIT DEV$V_DIR,DAP$V_DEVDIR : Map DIR bit
0137 263 $MAPBIT DEV$V_SDI,DAP$V_DEVSDI : Map SDI bit
013F 264 $MAPBIT DEV$V_SQD,DAP$V_DEVSQD : Map SQD bit
0147 265 $MAPBIT DEV$V_SPL,DAP$V_DEVSPL : Map SPL bit
014F 266 $MAPBIT DEV$V_NET,DAP$V_DEVNET : Map NET bit
0157 267 $MAPBIT DEV$V_FOD,DAP$V_DEVFOD : Map FOD bit
015F 268 $MAPBIT DEV$V_SHR,DAP$V_DEVSHR : Map SHR bit
0167 269 $MAPBIT DEV$V_GEN,DAP$V_DEVGEN : Map GEN bit
016F 270 $MAPBIT DEV$V_AVL,DAP$V_DEVAVL : Map AVL bit
0177 271 $MAPBIT DEV$V_MNT,DAP$V_DEVMNT : Map MNT bit
017F 272 $MAPBIT DEV$V_MBX,DAP$V_DEVMBX : Map MBX bit
0187 273 $MAPBIT DEV$V_DMT,DAP$V_DEVDMT : Map DMT bit
018F 274 $MAPBIT DEV$V_ELG,DAP$V_DEVELG : Map ELG bit
0197 275 $MAPBIT DEV$V_ALL,DAP$V_DEVALL : Map ALL bit
019F 276 $MAPBIT DEV$V_FOR,DAP$V_DEVFOR : Map FOR bit
01A7 277 $MAPBIT DEV$V_SWL,DAP$V_DEVSWL : Map SWL bit
01AF 278 $MAPBIT DEV$V_IDV,DAP$V_DEVIDV : Map IDV bit
01B7 279 $MAPBIT DEV$V_ODV,DAP$V_DEVODV : Map ODV bit
01BF 280 $MAPBIT DEV$V_RND,DAP$V_DEVRND : Map RND bit
01C7 281 $MAPBIT DEV$V_RTM,DAP$V_DEVRTM : Map RTM bit
01CF 282 $MAPBIT DEV$V_RCK,DAP$V_DEVRCK : Map RCK bit
01D7 283 $MAPBIT DEV$V_WCK,DAP$V_DEVWCK : Map WCK bit
51 52 D0 01DF 284 MOVL R2,R1 : Move data to correct register
FE1B' 30 01E2 285 BSBW FALS$CVT_BN4_EXT : Store DEV as an extensible field
01E5 286
01E5 287 :+
01E5 288 : Store fields derived from the File Header Characteristics XAB.
01E5 289 :-
01E5 290
22 56 10 E1 01E5 291 190$: BBC #DAP$V_LRL,R6,200$ : LRL implies 5 FHC fields (shortcut)
57 02F4 C8 DE 01E9 292 MOVAL FALS$L_FHCXAB(R8),R7 : Get address of FHCXAB
83 0A A7 B0 01EE 293 MOVW XAB$W_LRL(R7),(R3)+ : Send LRL
51 0C A7 D0 01F2 294 MOVL XAB$L_HBK(R7),R1 : Get HBK value
FE07' 30 01F6 295 BSBW FALS$CVT_BN4_IMG : Store HBK as an image field
51 10 A7 D0 01F9 296 MOVL XAB$L_EBK(R7),R1 : Get EBK value
FE00' 30 01FD 297 BSBW FALS$CVT_BN4_IMG : Store EBK as an image field
83 14 A7 B0 0200 298 MOVW XAB$W_FFB(R7),(R3)+ : Send FFB
51 28 A7 D0 0204 299 MOVL XAB$L_SBN(R7),R1 : Get SBN value
FDF5' 30 0208 300 BSBW FALS$CVT_BN4_IMG : Store SBN as an image field
FDF2' 30 020B 301 200$: BSBW FALS$BUICD_TAIL : Finish building message
05 020E 302 RSB : Exit

```

```

0000020F 304      .SBTTL FALSENCODE_NAM
020F 305      .PSECT FALSECODE      NOSHR,EXE,RD,NOWRT,BYTE
020F 306
020F 307      :++
020F 308      : Functional Description:
020F 309      :
020F 310      : FALSENCODE_NAM builds the DAP (resultant) Name message.
020F 311      :
020F 312      : FALSENCODE_NAM1 builds a DAP Name message as directed by input
020F 313      : parameters.
020F 314      :
020F 315      : Calling Sequence:
020F 316      :
020F 317      : BSBW FALSENCODE_NAM
020F 318      : BSBW FALSENCODE_NAM1
020F 319      :
020F 320      : Input Parameters:
020F 321      :
020F 322      : R8 Address of FAL work area
020F 323      : R9 Address of DAP control block
020F 324      : R10 Address of FAB
020F 325      : R11 Address of RAB
020F 326      :
020F 327      : And for FALSENCODE_NAM1 only:
020F 328      :
020F 329      : R5 Name type value
020F 330      : R6 Size of name string to use
020F 331      : R7 Address of name string to use
020F 332      :
020F 333      : Implicit Inputs:
020F 334      :
020F 335      : For FALSENCODE_NAM only:
020F 336      :
020F 337      : FAB$$_NAM
020F 338      : NAM$$_RSL
020F 339      : NAM$$_RSA
020F 340      :
020F 341      : Output Parameters:
020F 342      :
020F 343      : R0-R7 Destroyed
020F 344      :
020F 345      : Implicit Outputs:
020F 346      :
020F 347      : None
020F 348      :
020F 349      : Completion Codes:
020F 350      :
020F 351      : None
020F 352      :
020F 353      : Side Effects:
020F 354      :
020F 355      : None
020F 356      :
020F 357      :--
020F 358      :
020F 359      : FALSENCODE_NAM::
020F 360      : MOVZBL #DAP$M_FILSPEC,R5 ; Control point
55 01 9A 020F 360      : ; Get name type parameter

```


FALBLDATT
Symbol table

- BUILD DAP ATTRIBUTES MESSAGE

F 13

16-SEP-1984 01:38:03 VAX/VMS Macro V04-00
5-SEP-1984 01:16:27 [FAL.SRC]FALBLDATT.MAR;1

Page 9
(4)

DAPSB_ACCFUNC 00000040
DAPSB_ACCOPT 00000041
DAPSB_BITCNT 00000035
DAPSB_BKS 00000050
DAPSB_BSZ 00000052
DAPSB_DATATYPE 00000044
DAPSB_DCODE_FID 00000019
DAPSB_DCODE_MAC 00000018
DAPSB_DCODE_MSG 0000001A
DAPSB_FAC 00000042
DAPSB_FLAGS 00000031
DAPSB_FSZ 00000051
DAPSB_LEN256 00000034
DAPSB_LENGTH 00000033
DAPSB_NAME TYPE 00000040
DAPSB_ORG 00000045
DAPSB_RAT 00000047
DAPSB_RFM 00000046
DAPSB_SHR 00000043
DAPSB_STREAMID 00000032
DAPSB_TYPE 00000030
DAPSB_X_FIELD 00000024
DAPSK_BCN 000000C0
DAPSK_ATT_MSG = 00000002
DAPSK_BLN = 000000C0
DAPSK_BLS_D = 00000200
DAPSK_CREATE = 00000002
DAPSK_DATATYP_D = 00000002
DAPSK_DIR_LIST = 00000006
DAPSK_FIX = 00000001
DAPSK_NAM_MSG = 0000000F
DAPSK_SEQ = 00000000
DAPSL_ALQ1 0000004C
DAPSL_ATTMENU 00000040
DAPSL_CMWA 00000030
DAPSL_CRC_RSLT 00000020
DAPSL_DCODE_STS 00000018
DAPSL_DEV 00000068
DAPSL_EBK 00000078
DAPSL_FOP1 00000064
DAPSL_HBK 00000074
DAPSL_MRN 00000058
DAPSL_MSG_MASK 0000001C
DAPSL_SBN 0000007C
DAPSL_SSPWA 00000080
DAPSL_TEMP 00000090
DAPSM_ALQ1 = 00000040
DAPSM_BITCNT = 00000008
DAPSM_CBT = 00800000
DAPSM_CMPFMT = 00000008
DAPSM_CTG = 00000080
DAPSM_DEQ1 = 00000800
DAPSM_DFTSPEC = 00000010
DAPSM_DMO = 00002000
DAPSM_DSP_3NAM = 00000200
DAPSM_EBK = 00040000
DAPSM_EMBEDDED = 00000010

DAPSM_FFB = 00080000
DAPSM_FILSPEC = 00000001
DAPSM_FOP1 = 00001000
DAPSM_GET = 00000002
DAPSM_GO NOGO = 00000010
DAPSM_HBR = 00020000
DAPSM_IMAGE = 00000002
DAPSM_LRL = 00010000
DAPSM_LSA = 00000040
DAPSM_MACY11 = 00000080
DAPSM_MRS = 00000020
DAPSM_MSE = 00000010
DAPSM_ORG = 00000002
DAPSM_RAT = 00000008
DAPSM_RCK = 0000800C
DAPSM_RFM = 00000004
DAPSM_SBN = 00100000
DAPSM_SEGMENT = 00000040
DAPSM_TMP1\$ = 00000020
DAPSM_TMP2\$ = 000000C0
DAPSM_TMP3\$ = 00020000
DAPSM_TMP4\$ = 01000000
DAPSM_TMP5\$ = F0000000
DAPSM_WCK = 00004000
DAPSM_ZERO = 00000080
DAPSQ_DCODE_FLG 00000000
DAPSQ_FILESPEC 00000044
DAPSQ_MSG_BUF1 00000008
DAPSQ_MSG_BUF2 00000010
DAPSQ_NAMESPEC 00000044
DAPSQ_PASSWORD 00000050
DAPSQ_RUNSYS 0000005C
DAPSQ_SYSPEC 00000038
DAPSV_BKS = 00000007
DAPSV_BLS = 00000004
DAPSV_CBT = 00000017
DAPSV_CTG = 00000007
DAPSV_DATATYPE = 00000000
DAPSV_DEV = 0000000E
DAPSV_DEVALL = 0000000C
DAPSV_DEVAVL = 00000010
DAPSV_DEVCCCL = 00000001
DAPSV_DEVDIR = 00000003
DAPSV_DEVDMT = 00000008
DAPSV_DEVELG = 00000011
DAPSV_DEVFOD = 00000007
DAPSV_DEVFOR = 00000017
DAPSV_DEVGEN = 00000019
DAPSV_DEVIDV = 0000000D
DAPSV_DEVMBX = 00000012
DAPSV_DEVMNT = 0000000A
DAPSV_DEVNET = 00000018
DAPSV_DEVODV = 0000000E
DAPSV_DEVRCK = 00000015
DAPSV_DEVREC = 00000000
DAPSV_DEVRND = 00000014
DAPSV_DEVRTM = 00000013

FAL
Sym

FAL
FAL
FAL
FAL
FAL
FAL
FAL
FAL
FAL
FAL
FAL
FAL
FAL
FAL
FAL
FAL
FIL
MSG
PRO
RAB
RAB
RAB
SEN
SYS

PSE

\$AB
FAL

Pha

Ini
Com
Pas
Sym
Pas
Sym
Pse
Crc
Ass

The
318
The
277
20

DAPSV_DEVSDI = 00000004
DAPSV_DEVSHR = 00000008
DAPSV_DEVSPL = 00000009
DAPSV_DEVSQD = 00000005
DAPSV_DEVSWL = 0000000F
DAPSV_DEVTRM = 00000002
DAPSV_DEVWCK = 00000016
DAPSV_FSZ = 00000008
DAPSV_GEQ_V54 = 00000023
DAPSV_GEQ_V56 = 00000024
DAPSV_GEQ_V70 = 00000026
DAPSV_LRL = 00000010
DAPSV_MRN = 00000009
DAPSV_RCK = 0000000F
DAPSV_VAXVMS = 00000034
DAPSV_WCK = 0000000E
DAPSW_BLS = 00000048
DAPSW_DEQ1 = 00000054
DAPSW_DISPLAY1 = 0000004C
DAPSW_FFB = 00000072
DAPSW_LRL = 00000070
DAPSW_MRS = 0000004A
DAPSW_PARTNER = 00000006
DAPSW_VERSION = 00000004
DEVSV_ALL = 00000017
DEVSV_AVL = 00000012
DEVSV_CCL = 00000001
DEVSV_DIR = 00000003
DEVSV_DMT = 00000015
DEVSV_ELG = 00000016
DEVSV_FOD = 0000000E
DEVSV_FOR = 00000018
DEVSV_GEN = 00000011
DEVSV_IDV = 0000001A
DEVSV_MBX = 00000014
DEVSV_MNT = 00000013
DEVSV_NET = 0000000D
DEVSV_ODV = 0000001B
DEVSV_RCK = 0000001E
DEVSV_REC = 00000000
DEVSV_RND = 0000001C
DEVSV_RTM = 0000001D
DEVSV_SDI = 00000004
DEVSV_SHR = 00000010
DEVSV_SPL = 00000006
DEVSV_SQD = 00000005
DEVSV_SWL = 00000019
DEVSV_TRM = 00000002
DEVSV_WCK = 0000001F
FABS_BKS = 0000003E
FABS_FSZ = 0000003F
FABS_ORG = 0000001D
FABS_RAT = 0000001E
FABS_RFM = 0000001F
FABSC_REL = 00000010
FABSC_SEQ = 00000000
FABSC_STM = 00000004

FABSC_UDF = 00000000
FABSC_VFC = 00000003
FABS_L_ALQ = 00000010
FABS_L_DEV = 00000040
FABS_L_FOP = 00000004
FABS_L_MRN = 00000038
FABS_V_CBT = 00000015
FABS_V_CTG = 00000014
FABS_V_RCK = 00000017
FABS_V_WCK = 00000009
FABS_W_BLS = 0000003C
FABS_W_DEQ = 00000014
FABS_W_MRS = 00000036
FALS_BUILD_HEAD = ***** X 02
FALS_BUILD_TAIL = ***** X 02
FALS_ACCFUNC = 000001F6
FALS_ACCOPT = 000001F5
FALS_DATATYPE = 000001F4
FALS_DISABLE = 00000006
FALS_ENABLE = 00000005
FALS_LOGGING = 00000004
FALS_MISCOPT = 00000007
FALS_RAC = 000001F7
FALS_RBK_CACHE = 00000012
FALS_RCVBUFIDX = 00000011
FALS_VALUE = 00000010
FALSCVT_BN4_EXT = ***** X 02
FALSCVT_BN4_IMG = ***** X 02
FALSC_WRKBLN = 00002000
FALS_ENCODE_ATT = 00000000 RG 02
FALS_ENCODE_NAM = 0000020F RG 02
FALS_ENCODE_NAM1 = 0000022C RG 02
FALS_WRKBCN = 00002000
FALS_L_ALLXAB = 00000C00
FALS_L_ALLXABINI = 00000074
FALS_L_CHAIN_NXT = 0000007C
FALS_L_DATXAB = 00000320
FALS_L_FAB = 00000200
FALS_L_FAB2 = 00000800
FALS_L_FHCXAB = 000002F4
FALS_L_FOP = 000001F8
FALS_L_KEYNAM = 00001C00
FALS_L_KEYXAB = 00001000
FALS_L_KEYXABINI = 00000078
FALS_L_NAM = 00000294
FALS_L_NAM2 = 00000850
FALS_L_NUMBER = 000001FC
FALS_L_PROXAB = 0000034C
FALS_L_RAB = 00000250
FALS_L_RCVBUF = 0000005C
FALS_L_RDTXAB = 000003B0
FALS_L_RMS_PTR = 0000006C
FALS_L_STB = 000000C0
FALS_L_SUMXAB = 000003A4
FALS_L_TEMP = 000003F4
FALS_L_USE_SC1 = 000000A8
FALS_L_USE_SC2 = 000000AC

Mac

\$2
\$2
TOT
698
The
MAC

FALBLDATT
Symbol table

- BUILD DAP ATTRIBUTES MESSAGE

H 13

16-SEP-1984 01:38:03 VAX/VMS Macro V04-00
5-SEP-1984 01:16:27 [FAL.SRC]FALBLDATT.MAR;1

Page 11
(4)

```

FALS_L_USE_VER      000000A4
FALSQ_BLD           00000050
FALSQ_DIRNAME      00000088
FALSQ_FALLOG       00000090
FALSQ_FLG          00000000
FALSQ_MBX          00000038
FALSQ_MBXIOSB      00000030
FALSQ_RCV          00000040
FALSQ_RCVIOSB      00000020
FALSQ_RMS          00000064
FALSQ_STATE_CTX    00000008
FALSQ_SYSNET       00000098
FALSQ_TEMP         000003F8
FALSQ_VOLNAME      00000080
FALSQ_XMT          00000048
FALSQ_XMTIOSB      00000028
FALST_DAP          00000100
FALST_DIRNAME      00001F00
FALST_EXPAND       00000500
FALST_EXPAND2      00000A00
FALST_FALLOG       00001C00
FALST_FILESPEC     00000400
FALST_FILESPEC2    00000900
FALST_KEYBUF       00000700
FALST_MBXBUF       00001980
FALST_PRTBUF1      00001A00
FALST_PRTBUF2      00001B00
FALST_RESULT       00000600
FALST_RESULT2      00000B00
FALST_SYSNET       00001D00
FALST_VOLNAME      00001E00
FALSW_NEWNAM       = 0000000B
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
NAMS_B_RSL         = 00000003
NAMS_L_RSA         = 00000004
XABSL_EBK          = 00000010
XABSL_HBK          = 0000000C
XABSL_SBN          = 00000028
XABSW_FFB          = 00000014
XABSW_LRL          = 0000000A
  
```

↑-----↑
! 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	00000240 (576.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
-----	-----	-----	-----
Initialization	29	00:00:00.05	00:00:01.53
Command processing	106	00:00:00.37	00:00:01.34
Pass 1	319	00:00:07.95	00:00:35.18
Symbol table sort	0	00:00:01.01	00:00:03.04
Pass 2	80	00:00:01.47	00:00:05.60
Symbol table output	35	00:00:00.15	00:00:00.81
Psect synopsis output	1	00:00:00.03	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	572	00:00:11.03	00:00:47.54

The working set limit was 1500 pages.
64978 bytes (127 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1083 non-local and 56 local symbols.
377 source lines were read in Pass 1, producing 14 object records in Pass 2.
22 pages of virtual memory were used to define 21 macros.

! Macro library statistics !

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[FAL.OBJ]FAL.MLB;1	9
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	18

1354 GETS were required to define 18 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:FALBLDATT/OBJ=OBJ\$:FALBLDATT MSRCS:FALBLDATT/UPDATE=(ENHS:FALBLDATT)+LIBS:FAL/LIB

