```
FFFFFFFFFFFFFFF     AAAAAAAAA     LLL
FFFFFFFFFFFFFFF     AAAAAAAAA     LLL
FFFFFFFFFFFFFFF     AAAAAAAAA     LLL
FFF              AAA         AAA  LLL
FFF              AAA         AAA  LLL
FFF              AAA         AAA  LLL
FFF              AAA         AAA  LLL
FFF              AAA         AAA  LLL
FFF              AAA         AAA  LLL
FFFFFFFFFFFF     AAA         AAA  LLL
FFFFFFFFFFFF     AAA         AAA  LLL
FFFFFFFFFFFF     AAA         AAA  LLL
FFF              AAAAAAAAAAAAAAA  LLL
FFF              AAAAAAAAAAAAAAA  LLL
FFF              AAAAAAAAAAAAAAA  LLL
FFF              AAA         AAA  LLL
FFF              AAA         AAA  LLL
FFF              AAA         AAA  LLL
FFF              AAA         AAA  LLLLLLLLLLLLLLL
FFF              AAA         AAA  LLLLLLLLLLLLLLL
FFF              AAA         AAA  LLLLLLLLLLLLLLL
```

FILEID**FALBLDATT

```
FFFFFFFFFF    AAAAAA    LL          BBBBBBBB    LL          DDDDDDDD     AAAAAA    TTTTTTTTTT  TTTTTTTTTT
FFFFFFFFFF    AAAAAA    LL          BBBBBBBB    LL          DDDDDDDD     AAAAAA    TTTTTTTTTT  TTTTTTTTTT
FF           AA    AA   LL          BB      BB  LL          DD      DD  AA    AA      TT          TT
FF           AA    AA   LL          BB      BB  LL          DD      DD  AA    AA      TT          TT
FF           AA    AA   LL          BB      BB  LL          DD      DD  AA    AA      TT          TT
FF           AA    AA   LL          BB      BB  LL          DD      DD  AA    AA      TT          TT
FFFFFFFF     AA    AA   LL          BBBBBBBB    LL          DD      DD  AA    AA      TT          TT
FFFFFFFF     AA    AA   LL          BBBBBBBB    LL          DD      DD  AA    AA      TT          TT
FF           AAAAAAAAAA LL          BB      BB  LL          DD      DD  AAAAAAAAAA      TT          TT
FF           AAAAAAAAAA LL          BB      BB  LL          DD      DD  AAAAAAAAAA      TT          TT
FF           AA    AA   LL          BB      BB  LL          DD      DD  AA    AA      TT          TT          ....
FF           AA    AA   LL          BB      BB  LL          DD      DD  AA    AA      TT          TT          ....
FF           AA    AA   LLLLLLLLLL  BBBBBBBB    LLLLLLLLLL  DDDDDDDD    AA    AA      TT          TT          ....
FF           AA    AA   LLLLLLLLLL  BBBBBBBB    LLLLLLLLLL  DDDDDDDD    AA    AA      TT          TT          ....

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
```

FAL
VAX

Pha

Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
649
The
377
22

Mac

_$2
_$2
TOT

135

The

MAC

Table of contents

**F

```
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 FAL$ENCODE_NAM to support rename operation
0000    45  ;               (to use NAM2 for 2nd resultant Name message).
0000    46  ;
0000    47  ;       V03-002 KRM0084         K Malik         23-Mar-1983
0000    48  ;               Add support for STMLF and STMCR formats.
0000    49  ;
0000    50  ;       V03-001 KRM0068         K Malik         23-Nov-1982
0000    51  ;               Modified FAL$ENCODE_NAM to support rename operation.
0000    52  ;
0000    53  ;--
```

```
                        .SBTTL  DECLARATIONS

;
; Include Files:
;

        $DAPPLGDEF                      ; Define DAP prologue symbols
        $DAPHDRDEF                      ; Define DAP message header
        $DAPATTDEF                      ; Define DAP Attributes message
        $DAPACCDEF                      ; Define DAP Access message
        $DAPNAMDEF                      ; Define DAP Name message
        $DEVDEF                         ; Define device characteristics symbols
        $FABDEF                         ; Define File Access Block symbols
        $FALWRKDEF                      ; Define FAL Work Area symbols
        $NAMDEF                         ; Define Name Block symbols
        $XABDEF                         ; Define symbols common to all XABs
        $XABFHCDEF                      ; Define File Header Char XAB symbols

;
; Macros:
;
;       None
;
; Equated Symbols:
;

        ASSUME  DAP$Q_DCODE_FLG EQ 0
        ASSUME  FAL$Q_FLG EQ 0

;
; Own Storage:
;
```

```
                     88          .SBTTL  FAL$ENCODE_ATT
                     89          .PSECT  FAL$CODE        NOSHR,EXE,RD,NOWRT,BYTE
                     90
                     91 ;++
                     92 ; Functional Description:
                     93 ;
                     94 ;       FAL$ENCODE_ATT builds the DAP Attributes message.
                     95 ;
                     96 ; Calling Sequence:
                     97 ;
                     98 ;       BSBW    FAL$ENCODE_ATT
                     99 ;
                    100 ; Input Parameters:
                    101 ;
                    102 ;       R8      Address of FAL work area
                    103 ;       R9      Address of DAP control block
                    104 ;       R10     Address of FAB
                    105 ;       R11     Address of RAB
                    106 ;
                    107 ; Implicit Inputs:
                    108 ;
                    109 ;       DAP$V_GEQ_V54
                    110 ;       DAP$V_GEQ_V56
                    111 ;       DAP$V_GEQ_V70
                    112 ;       DAP$V_VAXVMS
                    113 ;       FAB fields
                    114 ;       FAL$B_ACCFUNC
                    115 ;       FAL$B_DATATYPE
                    116 ;
                    117 ; Output Parameters:
                    118 ;
                    119 ;       R0-R7   Destroyed
                    120 ;
                    121 ; Implicit Outputs:
                    122 ;
                    123 ;       None
                    124 ;
                    125 ; Completion Codes:
                    126 ;
                    127 ;       None
                    128 ;
                    129 ; Side Effects:
                    130 ;
                    131 ;       None
                    132 ;
                    133 ;--
                    134
                    135 FAL$ENCODE_ATT::                            ; Control point
 50   02   D0       136         MOVL    #DAP$K_ATT_MSG,R0          ; Get message type value
    FFFA'  30       137         BSBW    FAL$BUILD_HEAD             ; Construct message header
                    138
                    139 ;+
                    140 ; Determine which fields to return as follows:
                    141 ;   (1) always send the ORG, RFM, RAT, MRS, ALQ, DEQ, and FOP fields.
                    142 ;   (2) send the DATATYPE and BLS fields only if they do not equal their
                    143 ;       default values as defined in the DAP specification.
                    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 speaks 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$M_ORG>!-          ; Always send ORG, RFM, RAT, MRS, ALQ
                     000D   155                    <DAP$M_RFM>!-            ;  DEQ, and FOP fields
                     000D   156                    <DAP$M_RAT>!-            ;
                     000D   157                    <DAP$M_MRS>!-            ;
                     000D   158                    <DAP$M_ALQ1>!-           ;
                     000D   159                    <DAP$M_DEQ1>!-           ;
                     000D   160                    <DAP$M_FOP1>!-           ;
                     000D   161                  0>,R1                      ;
           01F6 C8   91 000D   162          CMPB    FAL$B_ACCFUNC(R8),-      ; Default DATATYPE field on
                06   0011   163                  #DAP$K_DIR_LIST            ;  directory list function as
                0B   13 0012   164          BEQL    10$                      ;  FAL$B_ACCFUNC = 0
           01F4 C8   91 0014   165          CMPB    FAL$B_DATATYPE(R8),-     ; Branch if DATATYPE = default value
                02   0018   166                  #DAP$K_DATATYP_D           ;
                04   13 0019   167          BEQL    10$                      ;
                001B   168          $SETBIT #DAP$V_DATATYPE,R1               ; Send DATATYPE
             3C AA   B1 001F   169 10$:     CMPW    FAB$W_BLS(R10),-         ; Branch if BLS = default value
           0200 8F   0022   170                  #DAP$K_BLS_D               ;
                04   13 0025   171          BEQL    20$                      ;
                0027   172          $SETBIT #DAP$V_BLS,R1                    ; Send BLS
        00   1D AA   91 002B   173 20$:     CMPB    FAB$B_ORG(R10),#FAB$C_SEQ
                04   13 002F   174          BEQL    30$                      ;
                0031   175          $SETBIT #DAP$V_BKS,R1                    ; Send BKS if org = REL or IDX
        10   1D AA   91 0035   176 30$:     CMPB    FAB$B_ORG(R10),#FAB$C_REL
                04   12 0039   177          BNEQ    40$                      ;
                003B   178          $SETBIT #DAP$V_MRN,R1                    ; Send MRN if ORG = REL
        03   1F AA   91 003F   179 40$:     CMPB    FAB$B_RFM(R10),#FAB$C_VFC
                04   12 0043   180          BNEQ    50$                      ;
                0045   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          BBC     #DAP$V_GEQ_V56,(R9),70$  ; Branch if partner uses DAP before V5.6
                0051   184 60$:     $SETBIT #DAP$V_DEV,R1                    ; Send DEV
           01F6 C8   91 0055   185 70$:     CMPB    FAL$B_ACCFUNC(R8),-      ; Branch if create function
                02   0059   186                  #DAP$K_CREATE              ;
                0B   13 005A   187          BEQL    80$                      ;  or
     07 69    23   E1 005C   188          BBC     #DAP$V_GEQ_V54,(R9),80$  ; Branch if partner uses DAP before V5.4
     51  001F0000 8F   C8 0060   189          BISL2   #<<DAP$M_LRL>!-          ; Send LRL, HBK, EBK, FFB, and SBN
                0067   190                    <DAP$M_HBK>!-            ;  fields
                0067   191                    <DAP$M_EBK>!-            ;
                0067   192                    <DAP$M_FFB>!-            ;
                0067   193                    <DAP$M_SBN>!-            ;
                0067   194                  0>,R1                      ;
        56   51   D0 0067   195 80$:     MOVL    R1,R6                    ; Save the send field flags
           FF93'   30 006A   196          BSBW    FAL$CVT_BN4_EXT          ; Store ATTMENU as an extensible field
                006D   197
                006D   198 ;+
                006D   199 ; Now store the designated fields in the order specified by ATTMENU.
                006D   200 :-
                006D   201
```

```
                05 56    00    E1  006D   202              BBC     #DAP$V_DATATYPE,R6,100$ ;
           83   01F4 C8  90    0071   203              MOVB    FAL$B_DATATYPE(R8),(R3)+; Store DATATYPE field
             83   1D AA  90    0076   204 100$:        MOVB    FAB$B_ORG(R10),(R3)+    ; Store ORG field
             83   1F AA  90    007A   205              MOVB    FAB$B_RFM(R10),(R3)+    ; Store RFM field
             0A 69    26  E0    007E   206              BBS     #DAP$V_GEQ_V70,(R9),110$; Branch if partner uses DAP V7.0
                          0082   207                                                    ;  or greater
             04   1F AA  91    0082   208              CMPB    FAB$B_RFM(R10),#FAB$C_STM;Is this format valid for DAP
                    04    1B    0086   209              BLEQU   110$                    ;  version 6.0 or earlier?
             FF A3    00  90    0088   210              MOVB    #FAB$C_UDF,-1(R3)       ; If not, say format is undefined
             83   1E AA  90    008C   211 110$:        MOVB    FAB$B_RAT(R10),(R3)+    ; Store RAT field
             0A 69    34  E0    0090   212              BBS     #DAP$V_VAXVMS,(R9),120$ ; Branch if partner is VAX/VMS
             04   1F AA  91    0094   213              CMPB    FAB$B_RFM(R10),#FAB$C_STM;Branch if not stream format
                    04    12    0098   214              BNEQ    120$                    ;
             FF A3    10  90    009A   215              MOVB    #DAP$M_EMBEDDED,-1(R3)  ; If it is, say cc is embedded
             04 56    04  E1    009E   216 120$:        BBC     #DAP$V_BLS,R6,130$      ;
             83   3C AA  B0    00A2   217              MOVW    FAB$W_BLS(R10),(R3)+    ; Store BLS field
             83   36 AA  B0    00A6   218 130$:        MOVW    FAB$W_MRS(R10),(R3)+    ; Store MRS field
             51   10 AA  D0    00AA   219              MOVL    FAB$L_ALQ(R10),R1       ; Get ALQ value
                  FF4F'  30    00AE   220              BSBW    FAL$CVT_BN4_IMG         ; Store ALQ as an image field
             04 56    07  E1    00B1   221              BBC     #DAP$V_BKS,R6,140$      ;
             83   3E AA  90    00B5   222              MOVB    FAB$B_BKS(R10),(R3)+    ; Store BKS field
             04 56    08  E1    00B9   223 140$:        BBC     #DAP$V_FSZ,R6,150$      ;
             83   3F AA  90    00BD   224              MOVB    FAB$B_FSZ(R10),(R3)+    ; Store FSZ field
             07 56    09  E1    00C1   225 150$:        BBC     #DAP$V_MRN,R6,160$      ;
             51   38 AA  D0    00C5   226              MOVL    FAB$L_MRN(R10),R1       ; Get MRN value
                  FF34'  30    00C9   227              BSBW    FAL$CVT_BN4_IMG         ; Store MRN as an image field
             83   14 AA  B0    00CC   228 160$:        MOVW    FAB$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 $CREATE).
                          00D0   234 ;-
                          00D0   235
           52   01F8 C8    D0    00D0   236              MOVL    FAL$L_FOP(R8),R2        ; Get partner supplied FOP bits
      52   0080C080 8F     CA    00D5   237              BICL2   #<<DAP$M_CTG>!-         ; Clear bits that may have changed
                                 00DC   238                        <DAP$M_CBT>!-        ;
                                 00DC   239                        <DAP$M_RCK>!-        ;
                                 00DC   240                        <DAP$M_WCK>!-        ;
                                 00DC   241                        0>,R2                ;
             51    04 AA    D0    00DC   242              MOVL    FAB$L_FOP(R10),R1       ; Get FOP bits returned by RMS
                          00E0   243              $MAPBIT FAB$V_CTG,DAP$V_CTG     ; Map CTG bit
                          00E8   244              $MAPBIT FAB$V_RCK,DAP$V_RCK     ; Map RCK bit
                          00F0   245              $MAPBIT FAB$V_WCK,DAP$V_WCK     ; Map WCK bit
             08 69    34  E1    00F8   246              BBC     #DAP$V_VAXVMS,(R9),170$ ; Branch if partner is not VAX/VMS
                          00FC   247              $MAPBIT FAB$V_CBT,DAP$V_CBT     ; Map CBT bit
                51    52  D0    0104   248 170$:        MOVL    R2,R1                   ; Move data to correct register
                      FEF6'  30    0107   249              BSBW    FAL$CVT_BN4_EXT         ; Store FOP as an extensible field
                          010A   250
                          010A   251 ;+
                          010A   252 ; Map and store the DEV field.
                          010A   253 ;-
                          010A   254
             03 56    0E  E0    010A   255              BBS     #DAP$V_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$:        MOVL    FAB$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_DEVCCL  ; 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$V_DEVWCK  ; Map WCK bit
          51   52    D0   01DF   284          MOVL    R2,R1                   ; Move data to correct register
             FE1B'    30   01E2   285          BSBW    FAL$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   FAL$L_FHCXAB(R8),R7     ; Get address of FHCXAB
          83   0A A7   B0   01EE   293           MOVW    XAB$W_LRL(R7),(R3)+     ; Send LRL
          51   0C A7   D0   01F2   294           MOVL    XAB$L_HBK(R7),R1        ; Get HBK value
                FE07'    30   01F6   295           BSBW    FAL$CVT_BN4_IMG         ; Store HBK as an image field
          51   10 A7   D0   01F9   296           MOVL    XAB$L_EBK(R7),R1        ; Get EBK value
                FE00'    30   01FD   297           BSBW    FAL$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    FAL$CVT_BN4_IMG         ; Store SBN as an image field
                FDF2'    30   020B   301 200$:    BSBW    FAL$BUILD_TAIL          ; Finish building message
                         05   020E   302           RSB                             ; Exit
```

```
                     304          .SBTTL  FAL$ENCODE_NAM
0000020F             305          .PSECT  FAL$CODE        NOSHR,EXE,RD,NOWRT,BYTE
                     306
                     307 ;++
                     308 ; Functional Description:
                     309 ;
                     310 ;       FAL$ENCODE_NAM builds the DAP (resultant) Name message.
                     311 ;
                     312 ;       FAL$ENCODE_NAM1 builds a DAP Name message as directed by input
                     313 ;       parameters.
                     314 ;
                     315 ; Calling Sequence:
                     316 ;
                     317 ;       BSBW    FAL$ENCODE_NAM
                     318 ;       BSBW    FAL$ENCODE_NAM1
                     319 ;
                     320 ; Input Parameters:
                     321 ;
                     322 ;       R8      Address of FAL work area
                     323 ;       R9      Address of DAP control block
                     324 ;       R10     Address of FAB
                     325 ;       R11     Address of RAB
                     326 ;
                     327 ;       And for FAL$ENCODE_NAM1 only:
                     328 ;
                     329 ;       R5      Name type value
                     330 ;       R6      Size of name string to use
                     331 ;       R7      Address of name string to use
                     332 ;
                     333 ; Implicit Inputs:
                     334 ;
                     335 ;       For FAL$ENCODE_NAM only:
                     336 ;
                     337 ;       FAB$L_NAM
                     338 ;       NAM$B_RSL
                     339 ;       NAM$L_RSA
                     340 ;
                     341 ; Output Parameters:
                     342 ;
                     343 ;       R0-R7   Destroyed
                     344 ;
                     345 ; Implicit Outputs:
                     346 ;
                     347 ;       None
                     348 ;
                     349 ; Completion Codes:
                     350 ;
                     351 ;       None
                     352 ;
                     353 ; Side Effects:
                     354 ;
                     355 ;       None
                     356 ;
                     357 ;--
                     358
                     359 FAL$ENCODE_NAM::                            ; Control point
55   01   9A    020F 360          MOVZBL  #DAP$M_FILSPEC,R5          ; Get name type parameter
```

```
50    0294 C8    DE  0212   361          MOVAL   FAL$L_NAM(R8),R0                ; Get address of Name block
  09 68    0B    E1  0217   362          BBC     #FAL$V_NEWNAM,(R8),10$          ; If sending 2nd Name message (rename
50    0850 C8    DE  021B   363          MOVAL   FAL$L_NAM2(R8),R0               ;  operation) use address of NAM2 block
                  0220   364          $CLRBIT #FAL$V_NEWNAM,(R8)              ; Clear flag
  56    03 A0    9A  0224   365 10$:     MOVZBL  NAM$B_RSL(R0),R6                ; <R6,R0> = resultant name string
  57    04 A0    D0  0228   366          MOVL    NAM$L_RSA(R0),R7                ;  parameter
                  022C   367
                  022C   368 FAL$ENCODE_NAM1::                                  ; Control point (with R5-R7 setup)
      50    0F    D0  022C   369          MOVL    #DAP$K_NAM_MSG,R0               ; Get message type value
          FDCE'   30  022F   370          BSBW    FAL$BUILD_HEAD                  ; Construct message header
      83    55    90  0232   371          MOVB    R5,(R3)+                        ; Store name type as an extensible field
      83    56    90  0235   372          MOVB    R6,(R3)+                        ; Store filespec as an image field
63    67    56    28  0238   373          MOVC3   R6,(R7),(R3)                    ; Copy name string into message
          FDC1'   30  023C   374          BSBW    FAL$BUILD_TAIL                  ; Finish building message
                  05  023F   375          RSB                                     ; Exit
                      0240   376
                      0240   377          .END                                    ; End of module
```

```
DAP$B_ACCFUNC                   00000040
DAP$B_ACCOPT                    00000041
DAP$B_BITCNT                    00000035
DAP$B_BKS                       00000050
DAP$B_BSZ                       00000052
DAP$B_DATATYPE                  00000044
DAP$B_DCODE_FID                 00000019
DAP$B_DCODE_MAC                 0000001B
DAP$B_DCODE_MSG                 0000001A
DAP$B_FAC                       00000042
DAP$B_FLAGS                     00000031
DAP$B_FSZ                       00000051
DAP$B_LEN256                    00000034
DAP$B_LENGTH                    00000033
DAP$B_NAMETYPE                  00000040
DAP$B_ORG                       00000045
DAP$B_RAT                       00000047
DAP$B_RFM                       00000046
DAP$B_SHR                       00000043
DAP$B_STREAMID                  00000032
DAP$B_TYPE                      00000030
DAP$B_X_FIELD                   00000024
DAP$C_BLN                       000000C0
DAP$K_ATT_MSG                 = 00000002
DAP$K_BLN                       000000C0
DAP$K_BLS_D                   = 00000200
DAP$K_CREATE                  = 00000002
DAP$K_DATATYP_D               = 00000002
DAP$K_DIR_LIST                = 00000006
DAP$K_FIX                     = 00000001
DAP$K_NAM_MSG                 = 0000000F
DAP$K_SEQ                     = 00000000
DAP$L_ALQ1                      0000004C
DAP$L_ATTMENU                   00000040
DAP$L_CMWA                      00000030
DAP$L_CRC_RSLT                  00000020
DAP$L_DCODE_STS                 00000018
DAP$L_DEV                       00000068
DAP$L_EBK                       00000078
DAP$L_FOP1                      00000064
DAP$L_HBK                       00000074
DAP$L_MRN                       00000058
DAP$L_MSG_MASK                  0000001C
DAP$L_SBN                       0000007C
DAP$L_SSPWA                     00000080
DAP$L_TEMP                      00000090
DAP$M_ALQ1                    = 00000040
DAP$M_BITCNT                  = 00000008
DAP$M_CBT                     = 00800000
DAP$M_CMPFMT                  = 00000008
DAP$M_CTG                     = 00000080
DAP$M_DEQ1                    = 00000800
DAP$M_DFTSPEC                 = 00000010
DAP$M_DMO                     = 00002000
DAP$M_DSP_3NAM                = 00000200
DAP$M_EBK                     = 00040000
DAP$M_EMBEDDED                = 00000010
DAP$M_FFB                     = 00080000
DAP$M_FILSPEC                 = 00000001
DAP$M_FOP1                    = 00001000
DAP$M_GET                     = 00000002
DAP$M_GO_NOGO                 = 00000010
DAP$M_HBK                     = 00020000
DAP$M_IMAGE                   = 00000002
DAP$M_LRL                     = 00010000
DAP$M_LSA                     = 00000040
DAP$M_MACY11                  = 00000080
DAP$M_MRS                     = 00000020
DAP$M_MSE                     = 00000010
DAP$M_ORG                     = 00000002
DAP$M_RAT                     = 00000008
DAP$M_RCK                     = 00008000
DAP$M_RFM                     = 00000004
DAP$M_SBN                     = 00100000
DAP$M_SEGMENT                 = 00000040
DAP$M_TMP1$                   = 00000020
DAP$M_TMP2$                   = 000000C0
DAP$M_TMP3$                   = 00020000
DAP$M_TMP4$                   = 01000000
DAP$M_TMP5$                   = F0000000
DAP$M_WCK                     = 00004000
DAP$M_ZERO                    = 00000080
DAP$Q_DCODE_FLG                 00000000
DAP$Q_FILESPEC                  00000044
DAP$Q_MSG_BUF1                  00000008
DAP$Q_MSG_BUF2                  00000010
DAP$Q_NAMESPEC                  00000044
DAP$Q_PASSWORD                  00000050
DAP$Q_RUNSYS                    0000005C
DAP$Q_SYSPEC                    00000038
DAP$V_BKS                     = 00000007
DAP$V_BLS                     = 00000004
DAP$V_CBT                     = 00000017
DAP$V_CTG                     = 00000007
DAP$V_DATATYPE                = 00000000
DAP$V_DEV                     = 0000000E
DAP$V_DEVALL                  = 0000000C
DAP$V_DEVAVL                  = 00000010
DAP$V_DEVCCL                  = 00000001
DAP$V_DEVDIR                  = 00000003
DAP$V_DEVDMT                  = 0000000B
DAP$V_DEVELG                  = 00000011
DAP$V_DEVFOD                  = 00000007
DAP$V_DEVFOR                  = 00000017
DAP$V_DEVGEN                  = 00000019
DAP$V_DEVIDV                  = 0000000D
DAP$V_DEVMBX                  = 00000012
DAP$V_DEVMNT                  = 0000000A
DAP$V_DEVNET                  = 00000018
DAP$V_DEVODV                  = 0000000E
DAP$V_DEVRCK                  = 00000015
DAP$V_DEVREC                  = 00000000
DAP$V_DEVRND                  = 00000014
DAP$V_DEVRTM                  = 00000013
```

```
DAP$V_DEVSDI                    = 00000004
DAP$V_DEVSHR                    = 00000008
DAP$V_DEVSPL                    = 00000009
DAP$V_DEVSQD                    = 00000005
DAP$V_DEVSWL                    = 0000000F
DAP$V_DEVTRM                    = 00000002
DAP$V_DEVWCK                    = 00000016
DAP$V_FSZ                       = 00000008
DAP$V_GEQ_V54                   = 00000023
DAP$V_GEQ_V56                   = 00000024
DAP$V_GEQ_V70                   = 00000026
DAP$V_LRL                       = 00000010
DAP$V_MRN                       = 00000009
DAP$V_RCK                       = 0000000F
DAP$V_VAXVMS                    = 00000034
DAP$V_WCK                       = 0000000E
DAP$W_BLS                         00000048
DAP$W_DEQ1                        00000054
DAP$W_DISPLAY1                    0000004C
DAP$W_FFB                         00000072
DAP$W_LRL                         00000070
DAP$W_MRS                         0000004A
DAP$W_PARTNER                     00000006
DAP$W_VERSION                     00000004
DEV$V_ALL                       = 00000017
DEV$V_AVL                       = 00000012
DEV$V_CCL                       = 00000001
DEV$V_DIR                       = 00000003
DEV$V_DMT                       = 00000015
DEV$V_ELG                       = 00000016
DEV$V_FOD                       = 0000000E
DEV$V_FOR                       = 00000018
DEV$V_GEN                       = 00000011
DEV$V_IDV                       = 0000001A
DEV$V_MBX                       = 00000014
DEV$V_MNT                       = 00000013
DEV$V_NET                       = 0000000D
DEV$V_ODV                       = 0000001B
DEV$V_RCK                       = 0000001E
DEV$V_REC                       = 00000000
DEV$V_RND                       = 0000001C
DEV$V_RTM                       = 0000001D
DEV$V_SDI                       = 00000004
DEV$V_SHR                       = 00000010
DEV$V_SPL                       = 00000006
DEV$V_SQD                       = 00000005
DEV$V_SWL                       = 00000019
DEV$V_TRM                       = 00000002
DEV$V_WCK                       = 0000001F
FAB$B_BKS                       = 0000003E
FAB$B_FSZ                       = 0000003F
FAB$B_ORG                       = 0000001D
FAB$B_RAT                       = 0000001E
FAB$B_RFM                       = 0000001F
FAB$C_REL                       = 00000010
FAB$C_SEQ                       = 00000000
FAB$C_STM                       = 00000004
FAB$C_UDF                       = 00000000
FAB$C_VFC                       = 00000003
FAB$L_ALQ                       = 00000010
FAB$L_DEV                       = 00000040
FAB$L_FOP                       = 00000004
FAB$L_MRN                       = 00000038
FAB$V_CBT                       = 00000015
FAB$V_CTG                       = 00000014
FAB$V_RCK                       = 00000017
FAB$V_WCK                       = 00000009
FAB$W_BLS                       = 0000003C
FAB$W_DEQ                       = 00000014
FAB$W_MRS                       = 00000036
FAL$BUILD_HEAD                    ********    X   02
FAL$BUILD_TAIL                    ********    X   02
FAL$B_ACCFUNC                     000001F6
FAL$B_ACCOPT                      000001F5
FAL$B_DATATYPE                    000001F4
FAL$B_DISABLE                     00000006
FAL$B_ENABLE                      00000005
FAL$B_LOGGING                     00000004
FAL$B_MISCOPT                     00000007
FAL$B_RAC                         000001F7
FAL$B_RBK_CACHE                   00000012
FAL$B_RCVBUFIDX                   00000011
FAL$B_VALUE                       00000010
FAL$CVT_BN4_EXT                   ********    X   02
FAL$CVT_BN4_IMG                   ********    X   02
FAL$C_WRKBLN                      00002000
FAL$ENCODE_ATT                    00000000 RG     02
FAL$ENCODE_NAM                    0000020F RG     02
FAL$ENCODE_NAM1                   0000022C RG     02
FAL$K_WRKBLN                      00002000
FAL$L_ALLXAB                      00000C00
FAL$L_ALLXABINI                   00000074
FAL$L_CHAIN_NXT                   0000007C
FAL$L_DATXAB                      00000320
FAL$L_FAB                         00000200
FAL$L_FAB2                        00000800
FAL$L_FHCXAB                      000002F4
FAL$L_FOP                         000001F8
FAL$L_KEYNAM                      00001C00
FAL$L_KEYXAB                      00001000
FAL$L_KEYXABINI                   00000078
FAL$L_NAM                         00000294
FAL$L_NAM2                        00000850
FAL$L_NUMBER                      000001FC
FAL$L_PROXAB                      0000034C
FAL$L_RAB                         00000250
FAL$L_RCVBUF                      0000005C
FAL$L_RDTXAB                      000003B0
FAL$L_RMS_PTR                     0000006C
FAL$L_STB                         000000C0
FAL$L_SUMXAB                      000003A4
FAL$L_TEMP                        000003F4
FAL$L_USE_SC1                     000000A8
FAL$L_USE_SC2                     000000AC
```

```
FAL$L_USE_VER                   000000A4
FAL$Q_BLD                       00000050
FAL$Q_DIRNAME                   00000088
FAL$Q_FALLOG                    00000090
FAL$Q_FLG                       00000000
FAL$Q_MBX                       00000038
FAL$Q_MBXIOSB                   00000030
FAL$Q_RCV                       00000040
FAL$Q_RCVIOSB                   00000020
FAL$Q_RMS                       00000064
FAL$Q_STATE_CTX                 00000008
FAL$Q_SYSNET                    00000098
FAL$Q_TEMP                      000003F8
FAL$Q_VOLNAME                   00000080
FAL$Q_XMT                       00000048
FAL$Q_XMTIOSB                   00000028
FAL$T_DAP                       00000100
FAL$T_DIRNAME                   00001F00
FAL$T_EXPAND                    00000500
FAL$T_EXPAND2                   00000A00
FAL$T_FALLOG                    00001C00
FAL$T_FILESPEC                  00000400
FAL$T_FILESPEC2                 00000900
FAL$T_KEYBUF                    00000700
FAL$T_MBXBUF                    00001980
FAL$T_PRTBUF1                   00001A00
FAL$T_PRTBUF2                   00001B00
FAL$T_RESULT                    00000600
FAL$T_RESULT2                   00000B00
FAL$T_SYSNET                    00001D00
FAL$T_VOLNAME                   00001E00
FAL$V_NEWNAM                  = 0000000B
FAL$W_DAPBUFSIZ                 0000001A
FAL$W_DISPLAY                   00000070
FAL$W_LNKCHN                    0000001C
FAL$W_MBXCHN                    0000001E
FAL$W_QIOBUFSIZ                 00000018
FAL$W_RECEIVED                  00000072
FAL$W_USE_DBS                   000000A0
FAL$W_USE_SYS                   000000A2
NAM$B_RSL                     = 00000003
NAM$L_RSA                     = 00000004
XAB$L_EBK                     = 00000010
XAB$L_HBK                     = 0000000C
XAB$L_SBN                     = 00000028
XAB$W_FFB                     = 00000014
XAB$W_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
ABS	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
_255DUA28:[FAL.OBJ]FAL.MLB;1	9
_255DUA28:[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 MSRC$:FALBLDATT/UPDATE=(ENH$:FALBLDATT)+LIB$:FAL/LIB
```

FALACTION
LIS

FALDAPCRC
LIS

FALMACROS
MAR

FALBLDXAB
LIS

FALBLDATT
LIS

FALBLDSTS
LIS

FALDEF
MDL

FALDAPIO
LIS

FALACTINI
LIS

FALACTMSG
LIS